PERSONALNI

računari

82

MAJ 1992.

CENA 1200 D

SOFTVER

BORLAND C++

OPERATIVNI SISTEMI

UNIX

NOVA NAGRADNA IGRA

SVE JE OVO VAŠE

COMTRAD

486 • 386 • 386SX

I JOŠ

30

VREDNIH NAGRADA



YU ISSN 0352-7271

9 770352 727009

PERSONALNI
računari 82

časopis za informatiku
i računarstvo
YU ISSN 0352-7221
Izdaje BIGZ
maj 1992. cena 1200 din.

Izdaje i štampa
Beogradski izdavačko-grafički zavod
11000 Beograd
Bulevar Vojvode Mišića 17
Generalni direktor
Ilija Rapajić
Direktor Novinskog sektora
Zoran Nikodijević
Glavni i odgovorni urednik
Jovan Regasek
Zamenik gl. i odg. urednika
Esad Jakupović
Tehnički urednik / sekretar redakcije
Natalija Desančić
Urednik za hardver
Dejan V. Veselinović
Stručna redakcija
Vesna Ćosić (aktuelnosti), Nenad Baločanin (baze podataka), Dejan Ristanović (softver), Jovan Skuljan (programiranje), Vladimir Stamenović (radne table), Dejan Veselinović (hardver), Dragutin Vuković (mreže), Zoran Životić (softver)
Adresa redakcije
11000 Beograd
Bulevar vojvode Mišića 17/III
Telefoni
653 748 (redakcija)
650-528 (prodaja)
Agencija BIGZ-a
(011) 651-783, 653-565
Teleks
11855 BIGZ YU
Telefaks
(011) 648-140, 647-656
Pretplata za zemlju
Za šest meseci 6000 (6 brojeva)
Za jednu godinu 11000 (11 brojeva)
(na žiro-račun D.P. BIGZ 60802-603-23264)
Pretplata za inostranstvo
Za jednu godinu 51 USD, 80 DEM, 67 CHS, 27 GBP, 271 FRF
(na devizni račun D.P. BIGZ 60811-620-16101-820701-999 03377)
Zbog usporenog prenošenja uplata preko banke, molimo pretplatnike da nam posle svake nove uplate odmah pošalju foto-kopiju uplatnice.
Rukopisi se ne vraćaju.

SEZAM
sistem za modemske komunikacije
(011) 648-899 (deset linija)
Upravnik sistema
Jovan Regasek
Administratori na sistemu
Zoran Životić i Dejan Ristanović
Moderatori
Dejan Balinde (CIVILIZACIJA), Jovan Bulajić (PC USER), Aleksandar Damnjanović (PC OS), Slobodan Kalezić (ORKA), Zoran Kehler (MOST), Vladimir Kostić (PC SOFT), Vladimir Kratonožić (VICEVI), Esad Jakupović (RAČUNARI), Delibor Lanik (ATARI ST), Pavle Peković (PC PROG), Dejan Ristanović (NOVOSTI i KOMUNIKACIJE), Bojan Tepavčević (FORUM), Dejan Veselinović (PC HARD) i Zoran Životić (SEZAM)
Sekretarica sistema
Natalija Desančić
Softver
Zoran Životić (glavna aplikacija) i Dejan Ristanović (eksterni programi)
Hardver
Server: COMPAQ DESKPRO 386/25, 3 MB RAM-a, tvrdi disk 300 MB (COMPAQ, Minhen). Mreža: dve stanice COMPAQ 286N/12, 1 MB RAM-a (COMPAQ, Minhen) i tri stanice „Ultra" AT 12 MHz, D-LINK DE 100 LanSmart („Ultra", Skopje). Modemi: VIVa (Thelema Communications, Beograd)

Sve o velikoj nagradnoj igri „Comtrad" & „Računari" na stranama 4 i 5

SADRŽAJ



Računari na naslovnoj strani razlikuju se od svojih prethodnika na istom prostoru po jednom važnom detalju: mogu da postanu vaši – besplatno! Firma "Comtrad" računare 486/33 MHz, 386/33 MHz i 386SX poklanja kao glavne nagrade u našoj velikoj igri koja počinje u ovom broju. Vi, da parafraziramo Minimaksa, narednih meseci sanjajte mašine, a mi ćemo Mariju Maćić, koju je, odevenu u elegantni poslovni komplet firme „Zmaj", ovekovečio Nenad Marjanović. Makeup je obezbedila Jelena Maćić.

junski broj časopisa računari izlazi iz štampe sredinom juna 1992.

Sve o velikoj nagradnoj igri „Comtrad" & „Računari" na stranama 4 i 5

Na osnovu Mišljenja Sekretarijata za informacije SR Srbije broj 413-01-40 od 4. 2. 1991. na ovaj časopis plaća se osnovni porez na promet po povlašćenoj tarifi od 3 odsto.

RAČUNARI NA POKLON
486 • 386 • 386SX

• Od ovog broja velika nagradna igra u četiri broja Računara" • Ukupno 33 lepe nagrade, od kojih tri izuzetne vrednosti • U svakom broju kupon sa tri pitanja • Javno izvlačenje 30. septembra

NAGRADE



I nagrada **COMTRAD 486**

33 MHz, 4 MB RAM, hard 105 MB Super VGA-monitor dva flopija, tastatura, tower kućište miš



II nagrada **COMTRAD 386**

33 MHz 4 MB RAM hard 105 MB Super VGA monitor, dva flopija tastatura tower kućište miš



III nagrada **COMTRAD 386SX**

20 MHz 1 MB RAM hard 40 MB Super VGA monitor, dva flopija tastatura baby tower kućište miš

Utešne nagrade

- 10 disk jedinica 3.5"/1,44 MB
- 10 miševa
- 10 kutija disketa

To još niste videli!

Jedinstvena prilika da najzad dobijete računar o kome ste sanjali!

Ne propustite sledeći broj „Računara" i svoju šansu da dobijete 486 – 386 – 386SX!

U saradnji sa uglednom firmom COMTRAD naš časopis pokreće nagradnu igru koja traje četiri meseca i čiji će rezultati biti objavljeni u oktobarskom broju „Računara". U igri će biti dodeljena ukupno 33 nagrade među kojima su tri vredna „Comtrad" računara – 486, 386 i 386SX.

Igra se odvija ovim redom:

- U ovom broju (81 – maj 1992) najava igre i prvi nagradni kupon sa pitanjima
- U sledećem broju (83 – jun 1992) prikaz prve nagrade (486) i drugi nagradni kupon sa pitanjima
- U broju 84 (jul/avgust 1992) prikaz druge nagrade (386) i treći nagradni kupon sa pitanjima
- U broju 85 (septembar 1992) prikaz treće nagrade (386SX) i četvrti nagradni kupon sa pitanjima
- U broju 86 (oktobar 1992) rezultati javnog izvlačenja (foto-reportaža)

Našim čitaocima firma „Comtrad" je dobro poznata. Pretpostavljamo da je ponajviše pamte po tome što je u našoj prošlogodišnjoj nagradnoj anketi, dosad najvećoj računarskoj nagradnoj igri na eks jugoslovenskim prostorima (sa oko 150 nagrada u hardveru, softveru i periferijskim uređajima) „Comtrad" bio darodavac glavne nagrade – jednog 386/33 računara. Neki „Comtrad" pamte po njegovom reklamnom nastupu u „Računarima" (redovno se oglašava već oko tri godine), neki po testovima hardvera na stranicama našeg časopisa. Ima mnogo i onih koji „Comtard" najbolje pamte po dobrom hardveru – mislimo, razume se, na zadovoljne vlasnike računara „Comtrad" (treba imati u vidu da je ova firma na našim prostorima do sada prodala oko 10 000 računara, sa odličnim rezultatima).

Šta je zapravo „Comtrad"?

"ComTrad" grupa je osnovana 1986. u Karakasu u Venecueli kao prodavnica IBM opreme. Već naredne godine otvorila je ured u Majamiju, a potom i u Diseldorfu. Godine 1988. osnovana je prodajna služba na Tajvanu, a kasnije i ured u Holandiji. Danas ova internacionalna grupa ima više od tridesel predstavništava u celom svetu, više stotina zaposlenih, vlastite pogone za proizvodnju računara, kao i veliku inženjersku službu za razvoj softvera, nekoliko hiljada dilera širom sveta, prodavnice u Venecueli i Španiji, a u Jugoslaviji filijalu u Beogradu.

Na eks-jugoslovenskim prostorima firma je prisutna već dve godine i sve je popularnija. Sedište joj je u Međunarodnom centru „Genex". Ima oko dvadeset zaposlenih, a radi prodaju gotovih sistema. Raspolaže velikim brojem dilera, o kojima se vodi dosta računa, tako da „Comtrad" i dileri nastupaju prema kupcima sa istom politikom. Neke dilere firma posebno ističe: „Infotrade", „Dugu", „Manir", „Iden" i „Comtrad Kragujevac". Treba naglasiti i da se servisi i garancije daju u Jugoslaviji. Sve isporuke i kontejnere firma planira šest meseci unapred, što je takođe rizik, jer je za to potrebno uraditi veoma kvalitetne procene o pravcima ulaganja, s obzirom da se tu radi o velikim količinama računara, a samim time i velikog kapitala.

U poslednje vreme „Comtrad" je na jugoslovenskom tržištu dosta snizio cene, imajući pre svega u vidu aktuelnu ekonomsku situaciju, odnosno smanjenu kupovnu moć pojedinaca i preduzeća. Takvo pojeftinjenje omogućeno je pre svega racionalizacijom proizvodnje i prodaje na evropskom prostoru.

Nedavno je, naime, „Comtrad" osnovao proizvodni pogon u Padovicama u Češkoslovačkoj. Veliki obim proizvodnje, korišćenje postojećih prednosti bivšeg istočnog bloka – jeftinije radne snage i blažih poreskih propisa – te sniženje cena transporta omogućili su firmi „Comtrad" da u dobroj meri smanji cena računara na čitavom evropskom tržištu, pa naravno, i jugoslovenskom. Na taj način ova firma sada konkuriše velikim proizvođačima na Tajvanu i u Americi. „Comtrad" je inače prva strana firma koja je prebacila deo proizvodnje sa Tajvana i iz Amerike u Istočnu Evropu. Na jugoslovenskom tržištu „Comtrad" je jedna od retkih firmi koja ima neograničene količine računara za promptnu isporuku.

"Comtrad" je dinamična firma, koja svoju delatnost širi na svim kontinentima, nastojeći da maksimalno zadovolji zahteve svojih kupaca. Zbog toga moto ove firme glasi: Your right choise! (Vaš pravi izbor)

Kontakt adresa

COMTRAD
11000 Beograd, V. Popovića 6
Poslovni centar Genex
Tel. 011/2224-151, 011/2222-652
Fax 011/2223-951

Veselin Jevrosimović, ComTrad



Dobra saradnja sa „Računarima"

"Comtrad" već više godina praktikuje nagradne igre u svim zemljama u kojima ima svoje institucije" – kaže gospodin Veselin Jevrosimović, suvlasnik firme, o motivima „ComTrad" da učestvuje u ovoj nagradnoj igri, sa nagradama znatne materijalne vrednosti. „Na taj način ostvaruje potpuniji kontakt sa tržištem i dolazi do kvalitetnijih informacija o popularnosti svojih proizvoda u svakoj od tih zemalja. Drugi motiv je činjenica da su „Računari" odigrali značajnu ulogu u afirmaciji „Comtrade". Tokom nekoliko godina dobre saradnje sa časopisom – kroz redovno oglašavanje, ustupanje hardvera na testiranje, učešće u velikoj nagradnoj anketi u prvoj polovini prošle godine – „Comtrad" je za čitalačku publiku ovog časopisa postao prepoznatljivo ime. Mi smo svesni da ovakve nagradne akcije imaju snažan marketinški efekat, odnosno da kao darodavac nagrada u uglednom časopisu kakav su „Računari" obezbeđujemo zapažen publicitet. Dodatni motiv je, svakako, i okolnost da je suvlasnik firme, čovek našeg porekla.

PROPOZICIJE

1. Pravo učešća, i mogućnost dobijanja neke od nagrada, imaju svi čitaoci „Računara". U nagradnoj igri ne mogu da učestvuju lica zaposlena u „Comtradu" i „Računarima" odnosno BIGZ-u, članovi njihovih porodica, kao ni lica koja su redovni saradnici ove dve firme.
2. Nagradna pitanja objavljuju se u četiri dela (sa četiri odgovarajuća kupona sa pitanjima): u „Računarima" broj 82 (maj 1992), broj 83 (jun 1992), broj 84 (jul/avgust 1992) i broj 85 (septembar 1992). Učesnici treba da štampanim slovima ili mašinom tačno odgovore na postavljena pitanja u sve četiri nagradna kupona.
3. Za dodelu nagrada u obzir će se uzimati jedino nagradni anketni kuponi iz „Računara".
4. Sva četiri nagradna kupona treba – isključivo zajedno! – dostaviti na adresu RAČUNARI, Bulevar vojvode Mišića 17, 11000 Beograd, sa obaveznom naznakom NAGRADNA IGRA. Rok za dostavu: 27. septembar 1991. (uvažava se poštanski žig).
5. Odgovori mogu da se šalju isključivo u standardnim kovertima manjeg ili srednjeg formata. U obzir za dodelu nagrada neće se uzimati veliki koverti (formata A-3 ili veći), koverti pravljeni u vlastitoj režiji, paketići i slično.
6. Redakcija „Računara" i „Comtrad", kao organizatori, ne mogu da snose nikakvu odgovornost za eventualni gubitak, oštećenje ili kašnjenje vašeg pisma u pošti.
7. Izvlačenje imena dobitnika nagrada izvršiće se javno 30. septembra 1992. godine. O eventualnom minimalnom pomeranju ovog datuma učesnici će biti obavešteni tokom trajanja nagradne igre.
8. Kompletni rezultati izvlačenja će biti objavljeni u oktobarskom broju (86) „Računara".

KUPON 1

Molimo da štampanim slovima upišete potrebne odgovore

1. Grupa "Comtrad" je osnovana ________ godine
2. Sedište "Comtreda" u Beogradu nalazi se u zgradi ________
3. Firma "Comtrad" svoje isporuke planira ________ meseci unapred

Ime i prezime ________

Ulica i broj ________

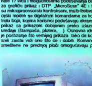
Pošt. broj i mesto ________

Kupon sačuvati i poslati na kraju igre zajedno sa preostalim kuponima na adresu RAČUNARI, Nagradna igra, Bulevar vojvode Mišića 17, 11000 Beograd

Monitori

ADI INTELIGENTNI MONITORI

Tajvanska korporacija ADI je predstavila novu liniju kolor monitora sa ekranom dijagonale 17 inča i mogućnostima podešavanja boja za grafički prikaz i DTP. „MicroScan" 4E i 5E su mikroprocesorski kontrolisani, multi-frekvencijski modeli sa digitalnim komandama za kontrolu boja, kojima korisnici podešavaju ekranski prikaz sa prikazom dobijenim preko izlaznih uređaja (štampača, plotera, ...). Osnovna ideja je postizanje što vernijeg prikaza, tako da korisnik zaista vidi ono što će i dobiti. Komande smeštene na prednjoj ploči omogućavaju po-



sebno podešavanje plave, zelene i crvene komponente signala, a mikroprocessor smešta podešene vrednosti u memoriju i postavlja ih svaki put kada se koristi aplikacija za koju su podešene. Postoji i reset taster koji daje mogućnost korišćenja fabrički podešenih nijansi. Pored tastera za kontrolu boje, tu su i kontrole za podešavanje horizontalne i vertikalne dimenzije slika, vertikalne pozicije i horizontalne faze za svaki displej mod posebno, a jednom podešene vrednosti se takođe posebno i memorišu.

Modeli 4E i 5E rade sa frekvencijom skeniranja do 64 KHz i raspolažu rezolucijom do 1280 x 1024 tačaka, a uz to se automatski prilagođavaju svim raspoloživim modovima. Oba modela su izrađena po svetskim sigurnosnim standardima i raspolažu univerzalnim izvorom napajanja koji radi sa naponima u rasponu od 90 do 264 V, veličina zrna fosforescentnog sloja modela 4E iznosi 0.31 mm, dok kod modela 5E iznosi 0.26 mm.

Kontakt fax: 886-2-713-6555.

Tržište

SEAGATE, DOS I WINDOWS

Ubuduće će se „Seagate" hard diskovi prodavati fabrički snabdeveni paketom *Windows 3.1* – to je rezultat dogovora koji su sklopili „Microsoft" i „Seagate" u cilju efikasnije promocije ovog operativnog sistema i olakšavanje posla sistem-dizajnerima i krajnjim korisnicima. *Windows* će biti instalirani tako da se sistem automatski budi pod prozorima, što je Alan Šugert, direktor „Seagate"-a nazvao „Instant-On" opcijom. Ugovor nije ekskluzivan, što znači da „Microsoft"-u ostavlja punu slobodu da sličan ugovor sklapa i sa drugim proizvođačima, a „Seagate"-u mogućnost da na svoje hard diskove instalira i neki drugi operativni sistem kao što je OS/2.

Fax kartice

BREAKOUT FAX 1.0 – KAKO POBEĆI OD STONOG FAKSA

Najkorisniji tip kompjuterskog faksa je svakako onaj koji omogućava ne samo slanje, već i primanje faksova kao pozadinski zadatak. Program koji ovo omogućava je *Breakout Fax 1.0*, a proizvodi ga firma „PC Communications", program radi pod DOS-om i podržava sve fax/modeme klase 2. Kad radi u pozadini *Breakout Fax* može samo da prima faksove, a u stanju je da presreće fajlove koji se šalju na štampač i, pod uslovom da su u Epson FX formatu, snima ih u fajl spreman za slanje faksom. Program zauzima 116 KB konvencionalne memorije, odnosno 74 KB ako se koristi expanded memorija.

Za sada program nije moguće koristiti iz *Windows*-a, koji mu ne dozvoljava da radi kao pozadinski program (verzija 3.0), što mu znatno umanjuje vrednost kao prijemnika za fax-poruka. Pod DOS-om program se može kontrolisati mišem ili tastaturom, dok pod *Windows*-ima može da mu se pristupa samo preko tastature. *Windows* verzija se očekuje tokom trećeg kvartala ove godine. Za korisnike koji rade pod DOS-om, *Breakout Fax* će biti veoma lak za upotrebu – sve opcije se biraju korišćenjem dijalog boksova, radio-dugmadi i sličnih objekata karakterističnih za grafičke radne sredine ali prilagođenih za rad u tekstualnom okruženju. Ipak, da bi se faks mogao i videti bilo da se prima ili šalje, treba imati grafički ekran, što je sređeno preko odvojenog programa *Faxview* čije zaduženje je briga o grafičkim elementima. Slika se prikazuje invertovana, kao belo na crnom, što predstavlja očigledan gest prema vlasnicima CGA monitora i kartica, ali korisnici koji raspolažu većim rezolucijama ne mogu ovo da srede.

Pri slanju faksova, program omogućava upotrebu elemenata iz pet različitih fajlova i njihovu sintezu u jedan faks, recimo zaglavlje, memo, potpis, fajl „skinut" direktno iz neke aplikacije i fajl sa diska koji može biti tekstualni fajl, bit-mapirana slika ili fajl u Epson FX formatu. Jeste da nije mnogo verovatno da će neko slati faks na ovaj način, ali mogućnost je tu. Svi faksovi se šalju sa datumom, brojem strane i brojem faksa u gornjoj liniji, a moguće je i umetanje kompletnog hedera, uključujući i logo. *Breakout Fax* sadrži i 16-to linijski editor koji omogućava brzo pisanje kratkih poruka ili notica uz dugačke fajlove.

Registrovanje i prijem dolazećih faksova se obavljaju automatski, pri čemu se smeštaju u fajlove nazvane tako da pokažu kad je faks prispeo – svaki naziv predstavlja skraćenicu meseca, dana, časa i minuta prispeća.

Kontakt adresa: Techland Systems International Ltd., Unit 4, Bourne End Business Centre Cores End Rd., Bourne End, Buckinghamshire SL8 5BB, U.K., 44-628-810022, fax 44-628-529116.

Mreže

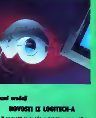
BEŽIČNA MREŽA ZA COMPAQ LTE

RangeLAN je naziv nova bežične mreže zasnovane na radio-talasima širokog spektra firme „Proxim", mreža je namenjena „Compaq" ovoj popularnoj LTE seriji prenosnih računara, prodaji će se naći krajem maja. Na raspolaganju se već nalaze ISA verzije *RangeLAN* adaptera za stone računare i servere. *RangeLAN/LT* predstavlja bežični mrežni adapter koji se postavlja u modemski slot „Compaq" LTE serije portabl računara i koji će biti opremljen istim setom drajvera kao i ISA mrežna kartica, što znači da će podržavati „NetWare" i „LANtastic" direktno, a „LAN Manager", „Banyan Vines", TCP/IP i druge mreže preko NDIS. Brzina prenosa podataka je deklarisana na 242 Kbps, slično brzini „AppleTalk" mreže, uz domet od oko 250 m u bilo kom okruženju koje ne sadrži metalne zidove.

Bežični prenos podataka unutar mreže i portabl računari predstavljaju gotovo idealne partnere – portabl računari opremljeni takvim adapterom se mogu prenositi sa jednog na drugo mesto i priključivati na lokalne mreže. Pošto *RangeLAN* takođe podržava tzv. *peer-to-peer* umrežavanje, moguće je i kreiranje „instant-mreže" među *RangeLAN* portablima u konferencijskoj sali ili nekom sličnom pogodnom mestu. Generalno gledano, što više primena pronađe, nova tehnologija će postajati sve vrednija. Budućnost *RangeLAN* sistema zavisi od razvoja prateće familije kartica koje će pokrivati razne tipove notebook računara. LTE, koji je drugi najpopularniji notebook računar, predstavlja idealnu odskočnu dasku.

RangeLAN/LT se neće prodavati uz računare, već će predstavljati samostalnu opciju namenjenu krajnjim korisnicima po ceni od $595.

Kontakt adresa: Proxim, Inc. 295 North Bernardo Ave., Mountain View, CA 94043 (415) 960-1630 Fax (415) 964-8181



Ulazni uređaji

NOVOSTI IZ LOGITECH-A

"Logitech" je pustio u prodaju novi softver za „ScanMan 256" ručni skener, kao i dva nova uređaja u svojoj liniji ergonomskih ulaznih uređaja. Reč je o novom modelu miša prilagođenom ljudima sa većim šakama nazvanog „MouseMan Large" i novoj verziji „TrackMan"-a, stacionarnog miša. Oba ove uređaja se isporučuju sa *MouseWare* utility paketom, u koji je uključen i specijalni *Windows* drajver za miševe sa tri tastera.

Dizajniran prema rezultatima studije većih šaka, „MouseMan Large" je za 10% veći od standardnog „MouseMan" modela, a novi „TrackMan" takođe predstavlja redizajniranu verziju već poznatog modela. Novi model koristi prirodnu zakrivljenost ruke i treba da omogući pun komfor i posle dužeg rada – dizajn je

posebno namenjen ljudima koji redovno koriste tekst-procesor sa mišem. Novi „TrackMan" će se praviti u dve verzije – serijskoj i bus.

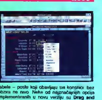
Novi softver za skener *FotoTouch*, omogućava tzv. *Virtual Page-Scanning* korišćenjem ručnog modela „ScanMan Model 256" i uključuje **Autostitch** funkciju koja omogućava automatsko spajanje dve, tri ili četiri skenirane trake u jednu homogenu celinu sa 256 sivih tonova. Upotrebom novog softvera postaje moguće skeniranja pune A4 pa čak i A3 strane ručnim skenerom umesto velikog i skupog flatbed modela. *FotoTouch* radi pod *Windows 3.0* ili novijim i omogućava direktno skeniranje u *Word for Windows* i *Ami Pro.*

Kontakt: Logitech SA, Switzerland, fax 41-21-859-9717

Windows scena

EXCEL 4.0 ZA WINDOWS

Istog dana kada se u prodaji pojavio *Windows 3.1*, „Microsoft" je lansirao i novu, odgovarajuću verziju Excel-a sa oznakom *4.0*. Verzija je označena kao *Major new release* i namenjena je, naravno, radu pod *Windows*-om *3.1*. Tokom razvoja su pažljivo analizirana iskustva i primedbe korisnika, tako da je akcenat stavljen prvenstveno na najčešće korišćene opcije. Tako je u *Excel 4.0 for Windows* ugrađeno čak 80% zahteva koje su tokom razvoja nove verzije korisnici uputili preko specijalne telefonske linije (nazvane „Microsoft WISH"). To se može primetiti već u bazičnoj upotrebi



tabele – poslu koji obavljaju svi korisnici bez obzira na nivo. Neke od najznačajnijih opcija implementiranih u novu verziju su **Drag and Drop** (omogućava pomeranje i kopiranje podataka jednostavnim potezom miša, što je već primenjeno u paketu *Word for Windows 2.0*), **AutoFill** (dozvoljava korisniku da unese više podataka u seriji, pri čemu ih sam program inteligentno prosleđuje), **Wizards** (napredna verzija Help-a koja vodi korisnika ka potpunom rešenju), **AutoFormat** (automatska primena jednog od 14 fabrički dizajniranih formata), **Shortcut Menus** (radi bržeg i direktnijeg pristupa često korišćenim komandama, pop-up meniji sadrže kontekst senzitivne komande bazirane na selektovanom objektu ili rasponu) i **Task-oriented Toolbars** (daju korisnicima jednostepeni pristup najčešće korišćenim komandama, za razne tipove poslova na raspolaganju stoje posebni **Toolbar**-ovi, a korisniku je omogućeno i pravljenje sopstvenih).

Naprednijim korisnicima na raspolaganju stoji set moćnih analitičkih funkcija za *What-If* analize, unakrsne proračune i kompleksne statističke, finansijske ili čisto numeričke proračune. Ove opcije su podeljene u nekoliko grupa. **Scenario Manager** eliminiše potrebu za snimanjem svakog seta pretpostavki u *What-If* analizama u poseban fajl – sada je omogućeno kreiranje i snimanje višestrukih *What If* scenarija u jedinstveni fajl, uz mogućnost kreiranja posebnih izveštaja. **Workbooks** predstavljaju ekstenziju koncepta radnog prostora primenjenog u verziji 3.0 i omogućavaju korisnicima da smeštaju povezane tabele u jedan fajl, čime se olakšava rad na projektima, kao i pravljenje sadržaja radi lakšeg pristupa, organizacije i podele fajlova. **Crosstabs** (*Common Tabulation Reports*) predstavljaju uobičajenu alatku za korisnika kojima treba analiza podataka iz baze radi identifikovanja trendova. **Analysis ToolPak** – u *Excel 4.0* je dodato 140 novih funkcija za rad sa kompleksnim brojevima i sigurnosne kalkulacije, kao i 20 novih procedura, uz već postojeći set finansijskih, inženjerskih i statističkih funkcija.

Pored svega navedenog na raspolaganju se nalazi i niz složenih ali ipak fleksibilnih funkcija za štampanja i prezentaciju, kao i podrška bivšim korisnicima paketa *Lotus 1-2-3*, kojima je posvećena posebna pažnja radi što bezbolnijeg prelaska, uključujući i poseban makro-interpreter koji omogućava upotrebu *1-2-3* makroa bez ikakvih izmena.

Paket se može dobiti kod svih ovlašćenih „Microsoftovih" distributera širom Evrope po ceni od 980 DEM, dok registrovani korisnici verzije 3.0 treba da doplate samo 295 DEM. Za rad je potrebno imati 286 ili jači sistem sa najmanje 2 MB RAM-a, 1.2-MB ili 1.44-MB disk-drajvom i „Microsoft" mišem.

DANI MS WINDOWS PROGRAMERA

POSTANI WINDOWS PROGRAMER

Kao deo tekućeg uzbuđenja koje okružuje predstavljanje i početak prodaje paketa *Windows 3.1*, „Microsoft" održava manifestaciju pod nazivom *Microsoft Windows Developers' Days* za Istočnu Evropu. Priredba je počela 1. aprila i trajaće do 31. jula, a projektovana je tako da omogući *Windows* programerima, bilo da je reč o početnicima ili iskusnim majstorima, da otkriju snagu i lakoću korišćenja „Microsoft"-ovih razvojnih proizvoda za *Windows*. Kao rezultat, „Microsoft" je privremeno snizio cene od 15-30% za *Visual Basic, Quick C for Windows* i *C++ 7.0*. Za registrovane korisnike „Microsoft"-ovih programerskih proizvoda takođe važi popust tokom četvoromesečne promocije, a prvi put u Istočnoj Evropi korisnici bilo koje verzije „Borland"-ovog C++ će moći da dobiju *upgrade* u vidu „Microsoft"-ovog C++ 7.0. Preporučene cene su snižene i za *QuickBasic*, a korisnici se animiraju da pređu na *Visual Basic*. Snižene cene važe samo za Istočnu Evropu uključujući i, kako se kaže u informaciji, Sloveniju, Hrvatsku i Jugoslaviju.

U okviru priredbe se održava i takmičenje za najbolji *utility* i najbolju igru napisane za *Windows* a sa ciljem da se pokaže kako je programiranje za *Windows* lako, naročito ako se koriste alati koje je napisao autor *Windows*-a lično. Pobednici će dobiti C++ 7.0 (prvo mesto), *Quick C for Windows* (drugo mesto) i *Visual Basic* (treće mesto). Uslovi za učešće na takmičenju su sledeći:



1. Izvorni kod i svi izvršni fajlovi će biti predati „Microsoft"-u kao *public domain*, što programer mora da potvrdi u pismenoj formi.
2. Programer mora da bude vlasnik izvornog koda, što takođe treba pismeno potvrditi.
3. Izvorni kod mora da bude dokumentovan.
4. Program može biti igra, *file-finder*, program za pretraživanje teksta ili bilo koji drugi korisni program koji radi pod *Windows* okruženjem.
5. Sve priloge treba slati na adresu: *Microsoft GmbH, att. Martin Dobes, Edisonstrasse 4, 8044 Unterschleissheim, Germany* najdalje do 30. juna 1992.
6. Program mora da bude napisan na lokalnom jeziku.
7. Sva prava zadržava „Microsoft".

Za detaljnije informacije o takmičenju se treba obratiti „Microsoft Partner"-ima u matičnoj oblasti.

Specijalne cene

CENE ZA NOVE KORISNIKE

Paket	Spec. cena	Reg. cena
C++ 7.0	590 DEM	980 DEM
QuickC for Windows, 1.0	345 DEM	395 DEM
Visual Basic 1.0	265 DEM	395 DEM
Quick BASIC, 4.5	165 DEM	195 DEM

CENE ZA REGISTROVANE KORISNIKE

Paket	Spec. cene	Reg. cena
sa C 6.0 na C++ 7.0	180 DEM	280 DEM
sa Quick C for DOS na Quick C for Windows, 1.0	155 DEM	175 DEM
sa QuickBasic ili MS BASIC na Visual Basic 1.0	100 DEM	- DEM
sa QuickBasic na QuickBasic 4.5	75 DEM	100 DEM
sa Borland C++ na MS C++ 7.0	360 DEM	980 DEM

Portabli

SLIMNOTE

"SlimNote" je naziv nove serije notebook računara tajvanske firme „Twinhead International" kojima je zajednička debljina od samo 42 mm i težina od 2 5 kg Serija se sastoji od četiri modela – najmanji početni model je „SlimNote-SX" baziran na AMD-ovom 386SX/25 sledeći model, „SlimNote-SL" koristi „Intel"-ov 386SL/25 sa 64 K keša Oba 386 modela su standardno opremljena sa 2 MB RAM-a, što se može proširiti do 4 ili 6 MB Za one koji traže nešto jače tu je „SlimNote-420" koji sadrži 486SX/20, dok je onima kojima ni ovo nije dovoljno namenjen model „SlimNote-433" zasnovan na 486DX procesoru koji radi na 33 MHz Oba 486 modela su opremljena sa 4 MB RAM-a što se može proširiti do 8 ili 20 MB.

Svi „SlimNote" modeli su opremljeni ugrađenim 3 5" flopijem i internim hard diskom kapaciteta od 40 do 120 MB prosečnih vremena pristupa od 19 ms ili boljih Tu je i VGA grafika sa *paper white* LCD-om rezolucije 640 x 480 koji može da prikaže do 64 nijanse sivog sa posebnim akcentom na brzinu reakcije dovoljnu da se uhvate potezi miša ili trekbola „Slim Note" je u stanju da istovremeno pogoni i LCD i klasični eksterni monitor, što ga čini idealnim za poslovne prezentacije Opciono se može dodati interni fax/modem i integrisani trekbol I/O linija uključuje paralelni i serijski port kao i 100-pinski ekspanzioni slot, a tu su i priključak za eksterni monitor i PS/2 tastaturu i miša Svi modeli su opremljeni odgovarajućim rutinama za ekonomisanje potrošnjom naročito u slučajevima kada su priključeni spoljni monitor i tastatura

Kontakt. Twinhead International Corporation, Taiwan fax 886-2-917-2674/5

Nove tehnologije

NOVI VLSI ČIPOVI



LH72001 i LH72/LH73 su nazivi novih familije VLSI čipova koji kombinuju nekoliko funkcija. Visoka integracija dozvoljava skraćivanje puteva signala i time veću brzinu i malu potrošnju energije što je naročito korisno za sisteme namenjene radu na ograničenom prostoru i uz ograničenu količinu energije kao što su palmtop računari „LH72001 Embedded Engine 1" predstavlja prvi IBM kompatibilni PC/XT na jednom jedinom VLSI čipu i snabdeven je sa 7 interfejsa oko NEC V20 procesora (kompatibilnim sa Intel 8088 8/16-bitnim procesorom) Tu su standardni serijski i paralelni I/O, brojač/tajmer Intrapt-kontroler, DMA kontroler, klok generator i bus kontroler Interfejsi za tastaturu štampač i memorijska kartice kao i kontroler memorije su priključeni u vidu dodatnih ugrađenih kola LH72001 prima do 1 MB ROM-a i 16 MB RAM-a, radi na maksimalno 8 MHz i isporučuje se u 160-pinskom QFP pakovanju

"LH72/73 Embedded Engine 2" su napravljeni tako da se prilagođavaju specifičnim potrebama i zadatim aplikacijama LH72 serija se takođe zasniva na V20 procesoru dok je serija LH73 zasnovana na V30 verziji (kompatibilan sa 16/16-bitnim Intel 8086 čipom) Budući da raspolaže bezkanalnim tipom nizova kapija sa po 15 000 30 000 ili 50 000 linijskih kapija dostupnih za makroćelijsku integraciju LH72/73 serija je takođe kompatibilna sa „Sharp"-ovom LZ98 serijom i odgovarajućom bibliotekom, tako da je moguće koristiti WORK VIEW bazirane *SF CAD Gate Array* razvojni alat za dizajniranje logike i simulacije Serija raspolažu istim interfejsima kao i LH72001

Kontakt adresa. Sharp Electronics (Europe) GmbH Hamburg, Fax 040-230-764

Windows scena

GEDYS – PRVI UPDATE ZA WINDOWS 3.1

GEDYS Windows Tools File Mnagement je jedan od prvih programskih paketa koji je dobio *update* za *Windows 3 1* Deo namenjen radu sa fajlovima se sastoji od četiri alata *MultiCopy TextSearch, SystemInfo* i *HexEdit. Multicopy* omogućava višestruko kopiranje diskova u bilo kom formatu podržava brzi format, uzastopno numerisanje i rad u mreži *TextSearch* predstavlja brz i dobro opremljen program za pretraživanje tekstualnih fajlova za hard diskove i mreže *SystemInfo* služi za grafičko predstavljanje računarskim resursa kao što su kapacitet hard diska status glavne memorije i opterećenje procesora a *HexEdit* predstavlja prvi heksadecimalni *Windows* editor Cena paketa je 295 DEM za pojedinačnu verziju ili 895 DEM za mrežnu verziju za pet korisnika

Kontakt GEDYS Software Consulting GmbH, Germany fax 49-531-38011-52

Laserski štampači

PRVI POTPUNO NE-JAPANSKI LASER

Korejska firma „Samsung" je predstavila svoj novi laserski štampač koji ne sadrži nikakve tehnološke novosti izuzev podatka da koristi prvu mašinu proizvedenu van Japana, i to u istoj tabrici „Finale 8000" kako je štampač nazvan, raspolaže sa šest rezidentnih emulacija, PCL 4 i 5 PostScript (klon) HPGL/2 IBM XL24 i Epson FX-850 i štampa brzinom od 8 strane u minutu a opremljen je elektronikom za automatsko prepoznavanje i preklapanje emulacije Elektronika sadrži „Intel" i960 RISC procesor koji radi na 16 MHz i 2 MB SIMM RAM-a (proširivo do 10 MB)

Pored velikog broja emulacije, štampač sadrži i četrnaest rezidentnih bit-mapiranih fontova osam vektorskih (scalable) i 35 *PostScript* fontova se standardno isporučuju uz štampač, a u kompaniji razmišljaju i o ugradnji *Bitstream PostScript* ili monotip fontova *TrueType* rasterizator osigurava kompatibilnost sa *Windows 3 1* fontovima. Zahvaljujući ugrađenom RISC procesoru „Finale 8000" je u mogućnosti da izbaci 7 5 *PostScript* strane u minutu.

"Auto Interface Switching" opcije omogućava automatsko preklapanje između *AppleTalk* Centronics RJ-11C RS-232 i 422C interfejsa. Svi ovi interfejsi izuzev *AppleTalk*-a spadaju u standardnu opremu, dok u opcije spadaju još i FAX, „Ethernet" i „Token Ring" interfejsi Tokom leta se očekuje i pojava *ImageKit*-a opcije za grafički intenzivne aplikacije koja će uključivati PostScript Level 2 kompatibilnost, 600 dpi rezoluciju obrazovanje nijansi sivog i 4 MB SIMM module

Cena štampača iznosi 1995 USD a *ImageKit*-a dodatnih 995 USD

Kontakt adresa. Samsung Electronics America Inc. 301 Mayhill St Saddle Brook NJ 07662 (201) 712-4815 Fax (201) 587 9178

Modemi

UFO-MATE



Prateći savremene trendove manjih tanjih i lakših notebook računara, tajvanska firma „E Tech" je predstavila seriju džepnih fax-modema pod nazivom *UFOmate Series* težine svega 180 grama uključujući i bateriju Najjači model serije je P9696MX prvi dzepni fax-modem opremljen G3 9600 bps faks i V 32 9600 bps modemskim funkcijama. Kao modem P9696MX je kompatibilan sa V 32 V.22bis V 23 V 22, V 21 kao i Bell 212A/103 Pored toga opremljen je MNP2-5 protokolom V 42 i V 42bis čime se efektivna brzina povećava na 38 400 bps Kao faks, P9696MX je kompatibilan sa Grupom 3, V 29 i V 27 protokolima i opremljen je *QuickLink II* softverom i može i da šalje i da prima faksove

Kontakt fax 886-35-777-751

WINDOWS ZA PEN RAČUNARE

Windows for Pens je naziv „Microsoftovog" operativnog sistema za pen računare koji je zasnovan na sistemu *Windows 3 1* i koji je pušten u prodaju dan posle osnovnog paketa U isto vreme firme „MicroSlate" „Momenta" NCR i NEC su sinhronizovano najavile da će ovaj operativni sistem biti fabrički instaliran na njihovim pen modelima a preko 220 proizvođača hardvera i softvera su najavili podršku Ekstenzivni dizajn *Windows 3 1* je omogućio integraciju specijalnog seta pen ekstenzija kao što su „mastilo" (*ink*) „gestovi" (*gestures*) i prepoznavanje rukopisa, oblika i specijalnih simbola. *Windows for Pens* pruža preko 70 novih pen funkcija koje omogućavaju proizvođačima softvera da kreiraju nove snažnije pen aplikacije Reč je o operativnom sistemu koji kombinuje funkcionalnost sa punom *Windows* kompatibilnošću i pen računarima pruža potpuno prirodnu radnu okolinu

QUO VADIS, PROCESORE?

Trendovi razvoja procesora, kao centralnog dela računara, danas su manje jasni nego što su bili juče. Sve neumerenije ambicije proizvođača softvera prisiljevaju proizvođače hardvera da stalno iznalaze nova rešenja. Sve oči su, pri tom, uprte u proizvođače procesora. Brzina jedne platforme, međutim, ne zavisi samo od brzine njenog procesora - ubrzanja se mogu postići i na drugoj strani.

Dejan V. Veselinović

Današnji računari se još uvek u najvećoj meri prave po uzoru na Fon Nojmanov model sekvencijalne mašine. To znači da centralni procesor obavi prvo jednu, pa drugu i tako redom sve zahteve koji se pred njega postavljaju. Suština je u tome da se svaki zahtev posebno obrađuje, odnosno što se različiti zahtevi, ma koliko se preklapali, ne obrađuju paralelno.

Ne ulazeći ovom prilikom u polemiku oko prednosti paralelne arhitekture, ipak se mora primetiti da se „paralelni" pristup sve više javlja i u svakodnevnom životu sasvim prosečnih korisnika. Mali primer bi bile video kartice – neme video kartice, računajući i „Hercules" kompatibilne, koje ne koriste usluge nekakvog, ma kako skromnog procesora (o savršenijim VGA kartama, i onima sa propisnim procesorima, da i ne govorimo).

Međutim, glavni teret rada i dalje podnosi centralni procesor. Njega su proizvođači već ubrzali koliko su mogli koristeći tradicionalne tehnologije, kao što je ugradnja relativno velikih keš memorija, podizanje radnog takta i slično. I ma šta uradili, kada se dobro zamislimo, opet se suočavamo sa problemom brzine.

Taj problem nastaje iz tri glavna razloga. Prvo, programi sa kojima radimo se jedne strane stalno rastu po obimu, a sa druge strane, gube na pakovanju koda, pa im su veći, tim je kod „rastreseniji", to jest sa više praznog hoda (misli se pre svega na programe koji pledìraju na univerzalnost u smislu platformi, kao recimo *MS Windows*). Drugo, standardna sabirnica je i dalje ista kao što je bila 1984. godine, a procesori su danas oko 7-10 puta brži, pa za otprilike toliko duže moraju i da čekaju. I treće, periferije su postale daleko složenije od onda do sada, sa jedne strane nude daleko više, ali i traže mnogo više.

Ovo je situacija koja će se sasvim sigurno sve više pogoršavati. Uzmimo samo jak prodor multimedija u računarstvo: do sada, računari su se bavili manom lokalnom grafikom, stvorenom unutar njih samih, a sada se odjednom od njih očekuje da se istovremeno bave sa više izvora grafike, sa zvukom i najzad da sve to uspešno okupe na jednom mestu. U takvoj situaciji nema druge nego sve od početka ubrzavati.

BRZO, BRŽE, NAJBRŽE

Prvo mesto kome se treba posvetiti su sama periferija. Uz poznata ograničenja sabirnice, najlogičnije je razvijati brzinu lokalno, na samim uređajima – time se rasterećuje centralni procesor, ali i sabirnica, čija relativna sporost prestaje da bude takvo ograničenje kada se broj transakcija preko nje svede na jednu ili dve, bez čestog dodavanja.

Diskovi su već sada osposobljeni za mnogo veće brzine od onih na kojima rade. Odavno su već postali inteligentni, a sada se radi na tome da postanu duplo inteligentniji. Jednostavni 8-bitni procesori koji su se do sada koristili polako ustupaju mesto hibridnim 16/8-bitnim (16-bitnim iznutra, 8-bitnim ka spoljnom svetu) procesorima, a pre svega starom „Intel" 80C188 čipu. Ovo je razrađenije verzija, u smislu mašinske integracije na samom čipu, dobrog, starog procesora 8088, i to u CMOS verziji. Pored jačeg procesora, i radni takt je veći, umesto 8 i 10 MHz kao do sada, koristi se 12 MHz. Primeri diskova koji koriste ovu tehnologiju su novi modeli firme Fujitsu (300, 420 i 510 MB), svi sa vremenom slučajnog pristupa od 9 mS, što je dobrih 55-76% brže od tipičnih proizvoda drugih proizvođača. Za sada, ovo je rezervisano za veće i skuplje modele, ali ne treba sumnjati da će se ova tehnologija vrlo brzo, negde do kraja ove godine, naći i na manjim modelima od 100 MB.

Princip je jednostavan – ubrzavanjem diska skraćuje se vreme koje procesor „drnda na prazno" dok ne dobije tražene podatke. Ovo je univerzalni pristup. A ima i lepših rešenja. Firma ALR je razvila posebnu tehnologiju neposrednog prebacivanja podataka sa diska, mimo standardne sabirnice, pravo u sistemsku memoriju. Dakle, umesto putanje disk-sabirnica-memorija, sada se sve završava na skraćenoj putanji disk-memorija. Tehničko rešenje se oslanja na lokalnu (to jest nestandardnu) sabirnicu sa posebnim kontrolerom, da li on prihvata sve IDE diskove ili ne, za sada se ne zna.

Treće rešenje je upotreba inteligentnog keš kontrolera, čija je funkcija da u celosti iskoristi sve mogućnosti sabirnice i da osetno ubrza fizički pristup disku, čime se takođe postižu veoma dobri rezultati, uz ukupno ubrzanje rada sa diskom od 5 i više puta. Još jedna vrlina ovog rešenja je univerzalnost – svaki IDE disk se može povezati na ovaj kontroler. Mana je, zna se – cena.

Slično važi i za video podsistem. Brzina rada video čipova se udvostručila, pa i utrostručila tokom poslednjih 2-3 godine. Uprkos tome, iole jače grafičke aplikacije zahtevaju posebne video kartice sa pravim procesorima, najčešće iz porodice TI 340XX, na taktovima od 25 do 60 MHz. Problem kod ovih kartica se svodi na posebne vezne programe potrebne radi usaglašavanja lokalnog video procesora sa ostatkom računara i naravno programom sa kojim se radi; više puta je potpuno ubedljivo pokazano i dokazano da su ovakvi vezni programi bar isto toliko bitni za brzinu rada kao i elektronika.

MOTORNA SILA RAZVOJA

Međutim, sve ovo je još uvek nedovoljno za multimedijske aplikacije, kao trenutno najzahtevniji segment računarstva. Ako pretpostavimo kratku animaciju od 10 sekundi, koja zbog očekivanog kvaliteta zahteva 30 slika u sekundi, pri čemu čak i najjednostavnije slike zauzimaju nekih 50 KB prostora na disku, lako dobijamo 15 MB prostora potrebnog samo za slike. Realnije slike će imati po 100-300 KB, a to su već kapaciteti diska od 30 do 90 MB. Ako vam se ovo čini prevelikim, znajte da samo jedna bolja slika napravljena u *3D-Studiu* kao od šale zauzme i ceo megabajt; u tom kontekstu, trebao bi nam kapacitet od oko 300 MB samo za slike.

Zato je razvijen niz procesora koji se lokalno bave kompresijom i dekompresijom baš ove vrste podataka. Da bi se sa jedne strane dobila kompresija od 20:1 ili više, a sa druge strane izbegla preduga čekanja prilikom dekompresije, nije bilo drugog rešenja nego da se ceo posao poveri posebnim lokalnim i specijalnim procesorima. Trenutno važeći standard u toj oblasti se skraćeno zove JPEG (*Joint Photographic Experts Group*) i zadužen je za uspostavljanje osnovnih standarda u ovoj oblasti, kako i industrija i korisnici ne bi opet bili suočeni sa nizom ličnih „standarda", što nužno izaziva tržišni haos. Postoji i drugi standard, poznat pod skraćenicom MPEG (*Moving Pictures Experts Group*), koji se bavi animacijom u stvarnom vremenu, a treći poznatiji standard, koji se bavi telekonferencijama, potiče od poznate CCITT grupe (*Consultative Committee for International Telegraph and Telephone*).

Dakle, upravo video oblast trenutno zauzima najistaknutije mesto motorne sile razvoja. U izgledu je vezani red bar tri procesora – centralnog procesora, lokalnog video procesora i procesora za kompresiju i dekompresiju video podataka.

Malo računice. Pretpostavljajući nekakav „standardni" format slike, recimo 352×240 piksela, postoji 1 320 blokova piksela od 8 puta 8 piksela (MPEG standard). Izmena sadržaja ovakve slike zahteva 896 sabiranja i 1024 množenja uz upotrebu algoritma za množenje matrica. To se prevodi na oko 2,5 miliona aritmetičkih operacija po jednoj jedinoj slici! Dakle, ako pretpostavimo 30 slika u sekundi, što se smatra dovoljnim za kvalitetnu animaciju bez titranja, proizlazi da nam treba platforma sposobna da izvede 76 miliona naredbi u sekundi. I to sve u samo dve boje, a ako želimo da dodamo i boju, ova brojka raste za faktor 2-4, što je već u domenu super računara.

Dalje, ako bismo se pridržavali CCIR (*Consultative Committee for International Radio*) standarda 601, koji propisuje sliku u boji i rezoluciji od 720×480 piksela, naše potrebe bi opet porasle za oko 4 puta. I tako, malo po malo, doguramo do računarske snage od 1,2 milijarde naredbi u sekundi, malo više nego što nam sada sedi na stolu ili pod njim.

U SUSRET MULTIMEDIJU

Ima, međutim, i pozitivnih strana. Osnovna je ta što se veoma retko (u kontekstu ukupnog broja) dogodi da se menja baš cela slika, najčešće se menja sam deo slike. Zato i CCIR i MPEG standardi aktivno koriste upravo ovu činjenicu da obrade samo razlike, a ne celinu. Jasno je da se time dobija na brzini rada, odnosno da se isti posao može obaviti i na osetno slabijoj platformi.

Uprkos tome, iskustva pokazuju da anima-



Nije ni sve u glavnom procesoru. Ugradnjom procesora specijalne namene, koji obavljaju namenske poslove u sistemu, kao što su video ili disk interfeja, može se značajno povećati snaga jedne platforme

cija zahteva još jače centralne procesore jer verovali ili ne procesor 80486 na 33 MHz je jedva dovoljno brz da iznese jednu bolju animaciju. Istina, „Intel" je opet otpočeo sa proizvodnjom verzija na 50 MHz, ali ni to nije zadovoljavajuće rešenje. Kao alternativa tokom ove godine će nam biti ponuđena cela serija 486 (SX i DX) sa mogućnosti rada sa dvostrukim taktom – učestanost kristala za sam procesor interno i isto to podeljeno sa dva eksterno. Drugim rečima, procesor će interno raditi na 66 MHz, a sa ostatkom sistema će komunicirati na današnjih 33 MHz. „Intel" očekuje da će ovo dovesti do realnih ubrzanja od 50-75%, čime bi se ovakve platforme izvukle iz sive zone mogućnosti multimedijskih primena.

Naravno, „Intel" treba da prikaže i svoj novi procesor, pod radnim nazivom „P5", popularno poznat kao 80586. Ako se obistine očekivanja, ovo bi mogao biti 64-bitni procesor sa početnim radnim taktom od 50 MHz, a ako bude sreće, i on će raditi na dvostrukoj učestanosti, pa bi to značilo da interno radi na 100 MHz. Veća širina sabirnice plus veći takt je jednako procesoru koji radi 4-6 puta brže od recimo 80486 na 33 MHz.

Nezavisno od novih procesora opšte name, IBM i „Intel" su se poslednjih nekoliko godina veoma intenzivno bavili novim pristupom multimedijam, od koga se očekuje da osvane u našim računarima kao što nedavno osvanulo i stono izdavaštvo, uprkos razlici u složenosti. Tako su razvili tzv. AVK (*Audio/Video Kernel*) arhitekturu, koja u konačnom obliku treba da nama korisnicima, omogući raspolaganje ekvivalentom od jedne studijske digitalne konzole za proizvodnju programa. Procesor se sastoji od nekoliko specifičnih funkcijskih blokove koji su interno međupovezani i ako da zajedno izvršavaju do čak 6 naredbi odjednom, što u kontekstu performansi iznosi koliko i jedan RISC procesor prve generacije na taktu od 150 MHz!

Najzad, kartice sadrži 2 MB posvećene video memorije, koja je proširiva do 4 MB. Ovaj standard je za sada internog tipa, ali je ponuđen i drugima u vidu licence.

INTELIGENTNE PERIFERIJE

Ovaj ne mali zadatak je ostvaren, bar u prvobitnom obliku, pomoću nekoliko specijalizovanih procesora povezanih u sistem na jednoj jedinoj kartici. Prvi procesor je „Intelov" 82750PB (procesor piksela), čiji je zadatak kada primi komprimovanu sliku i bit-mapiranu sliku, da prizove iz interne memorije mikrokodirane algoritme za dekompresiju, nakon čega se isprepletani video i audio signali razdvajaju u jedan niz video i dva niza audio signala (potrebnih za stereo zvuk) – a sve to u hardveru. Drugi procesor je 82750DB (procesor prikaza), koji iz bit-mapirane slike izvlači potrebne parametre, pretvara ih u RGB format, a zatim, posredstvom jednog digitalno-analognog (D/A) konvertora signal pretvara u sliku na ekranu. Radi na taktu do 45 MHz, što je sasvim dovoljno za super VGA prikaze (rezolucija 800×600).

Očigledno, ovde se radi o procesorima posebne namene (*Application specific processors*), no iako su to zaista pravi pravcati procesori, ne može se govoriti o paralelnim arhitekturama. Naime, njihova primena je takva da oni faktički rasterećuju centralni procesor niza radnji, ali ga ni u kom slučaju ne zamenjuju, niti mogu raditi bez njega i niza usluga koje im on pruža.

Isto važi i za inteligentne disk kontrolere i oni samo rasterećuju, ali ni na koji način ne zamenjuju centralni procesor. Jednostavno mu skraćuju vreme čekanja, ništa više.

I tako dolazimo do interesantne situacije da

se centralni procesor fiktivno ubrzava ugradnjom inteligentnih periferija koje su inteligentne zato što koriste univerzalne ili specifične procesore. Situacija se svodi na sledeće – umesto da centralni procesor mora da čeka recimo 10 taktova da se dobiju podaci sa diska, on sada čeka 3-4 takta, odnosno posao se obavlja 2,5 3,3 puta brže. Ovim se otvara novo pitanje – ako nastavimo da ubrzavamo periferije na takav način, treba li nam uopšte da jako brzo razvijamo radne taktove procesora, kada se jasno vidi da sada koristimo samo mali deo njihovih stvarnih mogućnosti (a ostatak vremena oni naprosto čekaju)?

Podsećamo čitaoce da je firma „Altos" pre godinu i po dana kao svoju najjaču mašinu prodavala UNIX platformu sa procesorom 80386 na svega 20 MHz. Uprkos skromnom radnom taktu, ta mašina je bila natprosečno efikasna samo zato što su diskovi bili pod kontrolom posebnog procesora (80186) i zato što su terminali takođe bili pod kontrolom posebnog procesora (80286). To je omogućavalo daleko bolje stvarno iskorišćenje centralnog procesora no što je uobičajeno. Ako hoćete, to je elegantan pristup.

Ovakav pristup u praktičnom smislu (odnosno u svakodnevnom radu svih nas) više nije naučna fantastika, kao što je bio pre jedva godinu ili dve (vidi prilog). Srećom, tehnologija stalno pojeftinjuje, mada naš standard pada daleko brže, ali to je već naš problem. Poenta je da se treba dobro zamisliti pre nego što odlučite da zamenite pola računara, jer je ogromna verovatnoća da i ovo što imate jedva da je iskorišćeno kako treba.

POTREBA ZA MEMORIJOM

Ono što je zajedničko radu svih, ali bukvalno svih, procesora ikada napravljenih je da će uvek raditi brže sa više memorije. Zato je naš lokalni običaj zadovoljavanja minimumom potrebnim da bi se uopšte podigao sistem zapravo omča oko vrata koju smo sami nameknuli i sada je stežemo. U praksi smo i previše puta videli platforme sa procesorom 80486 i svega 1 MB sistemske memorije. Ili ljude koji se bave ljutim CAD-om, crtajući slike koje traže bar-bratu 12-16 MB sa svega 4 MB na ploči, uz duboko verovanje da imaju veoma jake mašine.

Što je još gore, niz prodavaca nema pojma o stvarnim sistemskim zahtevima raznih platformi u kontekstu potreba korisnika. Svojevremeno je jedan prodavac iskrivio lice kada smo mu rekli da našem kolegi iz lista treba svih 16 MB za testiranje jednog programa, uz kiseo komentar u stilu „Pa je l' taj jede memoriju?" A praksa je nakon svega nekoliko dana pokazala da ni to nije bog zna šta, mirne duše je moglo biti još toliko, uprkos količini ipak je došlo do prebacivanja i iščitavanja sa diska. Program je bio *AutoCAD*.

Svi novi procesori o kojima smo rekli nekoliko reči će takođe tražiti sopstvenu memoriju, samo što će sada ta ukupna memorija biti raspodeljena po računaru. Recimo, 6 MB za sistem, 4 MB za keširanje diska i 2 MB za video kartu, čas posla zaposliste 14 MB, a to je proširivo do 52 MB (32 MB sistem, 16 MB keš kontroler, 4 MB video karta). Otud one stare računarske maksime: memorije i brzine nikada nemate previše.

Niti se svi ti posebni procesori razvijaju tek tako. Uzmite samo nekoliko podataka koji su već danas stvarnost – OS/2 2.0 će tražiti minimum od 4 MB sistemske memorije, a IBM preporučuje i celih 6 MB za udoban rad. *Windows 3.0* će istina raditi i sa 1 MB, ali da biste iz njega isterali ono što već sada ima, realno vam treba 4 MB. Prognoze oko „Microsoftove" nove tehnologije (NT) 32-bitnih *Windows*-a naredne generacije se takođe kreću od 4 MB kao minimuma, a 8 MB se pominje kao granica komfora.

Svaki procesor najbrže pristupa memoriji bez koje ne može da radi. Dakle, najbolji način da ga ubrzate je da mu dodate memorije. To podjednako važi i za inteligentne karte, mada su tu naravno apsolutne količine manje, katkad i mnogo manje. Primera radi, većini korisnika na današnjem nivou programa je dovoljno da imaju 1 ili 2 MB na keš kontroleru i 1 MB na video karti (jer sa tim već dobijaju 33.000 boja u rezoluciji od 600×600), a to je već (4+2+1) 7 MB u sistemu. U narednoj iteraciji, ovo će porasti na 11-12 MB, a odatle – ko zna?

U praktičnom smislu, razvoj novih procesora ipak postavlja i pitanje u kakvu platformu treba ugraditi sve te novotarije? Po nama, platforme sa procesorom 80286 su sada na potpuno istom mestu kao i pokojni XT pre dve ili tri godine – definitivno stvar prošlosti, sem za jednostavne poslove, mahom oko obrade teksta. I trendovi cene se kreću u tom smeru – danas jedna SX ploča na 25 MHz košta otprilike dva puta više od ploče sa 286 na 16 MHz, što je odnos koji do sada nije viđen u apsolutnim relacijama (to jest, nikad bolji). Međutim, u kontekstu cene celog računara, ta razlika u ceni se svodi na oko 10%, što nije mnogo kada jeftinija varijanta daje tehnološki i životno mrtvu platformu, a skuplja vam omogućava da koristite i sisteme sutrašnjice, a ne samo jučerašnjice.

SVE VIŠE PROCESORA

I za kraj, pogledajmo malo kakvi su to sve novi procesori. Dosta veliki deo njih ne samo što nije nov, nego je poprilično star. Recimo, procesor 80C186 je u osnovnoj NMOS verziji prikazan i pušten u prodaju još 1982/1983. godine. Danas u maloprodaji košta ispod 20 USD, pa se ne treba čuditi širokoj upotrebi na manje osetljivim poslovima, kao recimo disk-ovima i kontrolerima. Ne bi nas ni najmanje čudilo da se danas-sutra upotrebe i drugi dobro poznati procesori, kao recimo 80286.

Stalno pojeftinjenje je dovelo do toga da video kartice sa procesorom „Texas Instruments" 34010 padnu na nivo cene oko dva puta veći od običnih VGA kartica. Istina, radi se o procesoru opšte namene, za razliku od 34020, koji je namenjen upravo video aplikacijama, ali je i skuplji. Pa ipak, video kartice sa ovim procesorom na 32 MHz su dogurale do tri puta veće cene od standardnih VGA kartica, što je veliki uspeh imajući u vidu njihovu početnu cenu, i po 10 puta veću od standarda. Ako je suditi po dosadašnjim trendovima, tokom ove godine te kartice će dalje padati po ceni i napredovati po brzini, sve veći broj njih podržava i komplementarni numerički koprocesor koji opet ubrzava rad sa CAD-om.

Dakle, ima osnova očekivati da će sa krajem ove godine, a svakako tokom naredne, sve veći broj računara namenjenih profesionalnijim potrebama isporučivati sa tri procesora – centralnim, disk keš procesorom i video procesorom. Recimo, jedna DTP stanica bi mogla raditi sa čak četiri procesora – „Intel" 80XXX kao centralnim i keš procesorom, TI 34XXX kao video procesorom i „Motorola" 68XXX ili NS 32XXX procesorom kao srcem laserskog štampača. Ako još imate i faks/modem karticu, onda ćete imati i poseban signalni procesor na njoj, a to je već pet procesora u jednom sistemu. Nije loše, još samo da ih malo „nabifiamo" memorijom, pa da im navijemo radni takt, pa...

Pitanje koje se sada postavlja pred još uvek mnogobrojne vlasnike platformi sa procesorom 80286 na 16 MHz je šta sad oni da rade? Uzmimo da materijalno nisu raspoloženi za zamenu cele platforme, ali da žele da bar koliko-toliko ostanu u trendu razvoja novih oblasti računarstva.

Iz svega gore pomenutog, nadamo se da je jasno da u svakom slučaju procesor provodi dosta vremena radeći u prazno, odnosno čekajući zatražene podatke. Dakle, ako nećete ili ne možete da zamenite celu matičnu ploču, prvo i osnovno što možete je da dodate malo memorije. Ako imate 1 MB, proširite ga na bar 2 MB, a ako je ikako moguće, i celih 4 MB. Ukoliko dogurate do 4 MB (a ne zaboravite, ta memorija se može koristiti i naknadno u novoj mašini), odvojite bar 1 MB za keširanje tvrdog diska, time ćete ubrzati rad bar dva do tri puta i sebi olakšati život.

SLIKA I ZVUK

Ako se spremate da kupite računar, dobro razmislite za šta ćete ga koristiti, lako bi se moglo desiti da za vaše potrebe bude daleko bolji posao kupiti jednu SX matičnu ploču na 25 MHz i 64 KB keša umesto 486 na 33 MHz, ali uz investiranje razlike u ceni u neki inteligentni keš kontroler sa memorijom. Evo jedne računice: razlika u ceni između pomenute SX ploče i 486 na 33 MHz iznosi oko 1.400 DEM, a pomenuti keš kontroler sa 4 MB RAM-a košta oko 900 DEM. Dakle, platforma sa SX-om i 6 MB sistemske memorije prođe sa inteligentnim kontrolerom i 4 MB je oko 100 DEM jeftinija od ploče sa 80486 na 33 MHz bez ikakve memorije. Međutim, ako ste programer, ili ako se bavite bazama podataka, jeftinija verzija će sasvim sigurno mnogo bolje zadovoljavati vaše potrebe od skuplje.

Problem postaje zaista nezgodan ako nameravate da se bavite računarskom animacijom. Tu dosta nema zamene za brz procesor, za brzu video kartu i za brz disk – drugim rečima, treba vam sve. U tom slučaju čeka vas račun od nekih DEM 6.500-7.000 u inostranstvu, i to kada štedite, pa uzmete „Tseng" ET4000 video karticu koja može da reprodukuje 33.000 boja, a ne neku karticu sa TI 34010/34020, koja bi objektivno podigla cenu sistema za bar 25%.

Spas vas očekuje kod diskova, koji će poslati sve inteligentniji i brži, pa kod video karti, koje postaju sve brže i bolje, i najzad, kod posebnih procesora za specijalne namene, koji će vas rasteretiti prevelikih mašinskih zahteva. Primera radi, procesor koji je razvila firma „C-Cube", sa oznakom CL 550B, se uklapa u JPEG standarde i omogućava kompresiju neposrednim skidanjem slike sa ekrana. IIT je izbacio na tržište čip pod imenom „Vision Processor", koji sadrži RISC procesor i 64 64-bitna registra. Tokovi podataka su regulisani veoma brzom logikom, a procesor je programabilan u mikrokodu, kako bi se lakše prilagodio različitim standardima. Na žalost, čip obrađuje samo 15 slika u sekundi.

Dosta nade se polaže u novi „Intelov" čip pod šifrom „750". Radi se o procesoru koji može da održi ritam od 30 slika u sekundi za kompresiju ili dekompresiju, ali ne oboje istovremeno. „Intel" navodno planira da iz osnovnog projekta razvije čitavu porodicu multimedijskih procesora, u kojoj će najjači član biti u stanju da pored slike obrađuje i zvuk. U osnovi, ideja je da ovaj čip, uz nekoliko veznih pomoćnih kola, zameni više od polovine sadržaja gore opisane AVK kartice, naravno, uz još malo brzine.

NESTAŠNI ČIKONI

Evo još jednog malog i simpatičnog prenosnika, ovoga puta poreklom od tajvanske firme „Chicony". U domenu prenosnika, „Chicony" model LT-3600 je i dan-danas naša referenca; bilo je interesantno pogledati kako se ova firma snašla sa novom generacijom mašina.

Dejan V. Veselinović

Nedavno smo prikazali dva mala prenosiva modela, jedan od firme „Daewoo", a drugi poznate američke firme „Dell". Evo prilike da pogledamo još jednu mašinu na sličan kalup, čiji je proizvođač tajvanska firma „Chicony".

Nekoliko reči o samoj firmi „Chicony". Ova firma je kod nas poznata po svojim tastaturama i raznim proizvodima za dogradnju računara, naročito I/O i VGA karticama. Međutim, malo se zna da je ona, zapravo, proizvođač u projektima daleko poznatijih i skupljih imena, kao što su američke firme „Zenith" i „Mitsuba" (da to je američka firma a naziv modela je „Ninja"). Oni daju projekte, a „Chicony" ih realizuje, dobro jednima, dobro drugima. Naravno, sa iskustvom stečenim na takav način, bilo je sasvim normalno da ova firma izbaci i neke modele pod sopstvenim imenom, tako se pojavila serija 3400/3500/3600, zatim model 5600 i sada ovaj najnoviji, model NB 5620.

LIČNI OPIS

Radi se o pravom „notebook" računaru sa dimenzijama 210×297×47 mm (DxŠxV) i težinom od jedva 3,2 kg za celu mašinu, računajući tvrdi disk i akumulator. Kućište je završno obrađeno „Nextel"-om, premazom koji daje blago hrapav osećaj, koji veoma podseća na somot. Boja je veoma poželjna tamno siva, gotovo crna.

Konstrukciju mašine bismo ocenili kao veoma dobru do odličnu, ne toliko zbog izuzetne čvrstoće i krutosti (obe osobine su na nivou standarda, ali ništa bolje), koliko zbog nekih drugih primenjenih rešenja. Akumulator, numerički koprocesor i memorijska proširenja se u najmanju ruku veoma lako i brzo ugrađuju zahvaljujući poklopcima na dnu računara. Prosto ih otvorite i sami ugradite šta god želite, odnosno zamenite akumulatore. Ovo je do sada najfunkcionalnije rešenje koje smo ikada videli, mada je bilo i drugih pokušaja na istu temu.

U vezi proširenja, i jedna zamerka. Naime, dopunska memorija, a to je sve preko 1 MB, nalazi se na posebnoj štampanoj pločici. Dakle, ako hoćete više memorije, vezani ste za proizvođača. U principu ne volimo ovakve rešenja, jer se prečesto događa da memoriju koju biste inače platili 100 jedinica ovako platite i po dva ili tri puta više (princip Henrija Forda – jeftin auto, skupi delovi). Smatramo da je ipak trebalo ići na standardne SIM module, kao i u starijem modelu 3600, uprkos razlici u dimenzijama.

Isto kao i na starijem modelu 3600, a naravno i konkurenciji pre svega u vidu modela „Dell" DN320, računar raspolaže ugrađenom flopi disk jedinicom od 3,5 inča i 1,44 MB, kao i tvrdim diskom od pristojnih 41 MB. Po svemu sudeći, taj disk je veoma nalik onome u DN320, jer im se krive brzine prenosa poklapaju do decimalnog mesta, mada su neki rezultati rada sa diskom različiti. Sama jedinica je dosta brza, ispod 15 ms, što nije za bacanje, pa je tu najveći pomak od starijeg modela 3600.

Prikaz koristi CCFT LCD tehnologiju, kao i stariji model i kao svi savremeni modeli, a to znači da ima pozadinski osvetljen LCD ekran u rezoluciji standardne VGA karte, odnosno 640×480. Video podsistem, po tradiciji kuće, koristi „Cirrus Logic" video čipove. Razlog je najverovatnije taj što ovi čipovi kao ni jedni drugi isteruju veliki broj nijansi na poslovično kilavim LCD ekranima.

Najzad, sama platforma koristi procesor 80386SX na 20 MHz, bez ikakvog keša, ali zahvaljujući proširenju od 1 MB (za ukupno 2 MB u sistemu), sa bar dvostrukim preplitanjem, uz odgovarajuće dobiti u performansama. Naravno, tu je i postolje za opcioni numerički koprocesor.

Tastatura i uopšte „komandna tabla", su posebna priča. Tastature na ovako smanjenim računarima su uvek sumnjive rabote, kako zbog potrebe da se emulira prava AT tastatura, zbog potrebnih funkcija, tako i zbog kratkog hoda tastera i njihove veličine, koji su uvek u funkciji dimenzija mašine. Do sada nismo videli brilijantno rešenje ovog problema, mada nam se kompromis napravljen na „Daewoo" mašini dopao. Potpuno isto važi i za ovu tastaturu, koja sa svojih 83 tastera pokušava da emulira 101 taster, veoma je nalik onoj na „Daewoo" mašini, pa iako je kompromis dobar, opet ovo ostaje kao najveći nedostatak ovakvih mašina. U poređenju sa starijim modelom 3600, moglo bi se reći da je ova tastatura poprilično zaostala u svakom pogledu. Zato je deset LED indikatora prava stvar, pre svega zbog niza informacija koje obezbeđuju od stanja uključenosti do porekla struje koju mašina troši i statusa tri tastera (Scroll, Caps i Num Lock). Svaka čast.

Kvalitet prikaza lako pokazuje napredak postignut tokom poslednje dve godine. Radi se o istina manjem, ali daleko kvalitetnijem prikazu. Iako još uvek ima „curenja" tu i tamo, to je sada ipak svedeno na izuzetke pre nego na normalnu pojavu. Komande za podešavanje nivoa osvetljenosti i kontrasta se nalaze na desnoj strani samog prikaza, rešenje koje se nama daleko više sviđa od drugih mesta, a posebno kućišta.

Najzad, na poleđini mašine se nalazi jedan mali poklopac, koji kada se otvori omogućava pristup raznim veznicima – za struju, za dve RS 232C linije, za jednu paralelnu liniju, za spoljni monitor, a tu je, hvala ti Bože, i poseban veznik za spoljnu disk jedinicu od 5,25 inča (koja je opciona). Utičnica za spoljnu tastaturu se nalazi sa leve strane kućišta. Za spoljnu tastaturu i spoljni disk ćete dobiti i potrebne vezne/prelazne kablove uz računar. Od opcione opreme, možete ugraditi numerički koprocesor (bilo čiji), možete dokupiti rezervni akumulator, spoljnu disketnu jedinicu od 5,25 inča, numeričku tastaturu i modem na 2 400 boda, najverovatnije u aranžmanu proizvođača.

ALI, ALI, ...

Lepo jeste, ali ima i zamerki, i to na žalost dosta. Prvo, nije nam jasno zašto memorija nema proveru pariteta (*parity check*). Da im nije možda bilo skupo, ili da nije ponestalo prostora? Najzad, radi se o samo još dva čipa (jedan na ploči, drugi na kartici).

Drugi problem se svodi na video sistem, on nije onako čist kao što bi trebalo da bude – najzad, takvi podsistemi se prave već godinama, a onaj u starijem modelu 3600 radi odlično. Na testovima kompatibilnosti smo naišli na previše problema, sinhronizacija sa časovnikom od 70 Hz ne radi, ima uobičajenih problema sa interaptom za vertikalno brisanje, a skrolovanje sa desnog donjeg u levi gornji ugao je veoma cimavo. Sve to manje-više, ali nakon svakog izlaska iz potprograma *COMPRESS* iz *PC-Tools* paketa 7.0 ceo ekran bi



Nova Chicony generacija: NB 5620

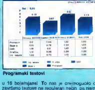
Snimio: Nenad Marijanović

Slika 1



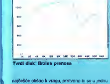
Mašinski testovi

Slika 2



Programski testovi

Slika 3



Tvrdi disk: Brzina prenosa

najčešće otišao k vragu, pretvorio bi se u jednu psihodeličnu sliku koju čisti samo RESET

Ne znamo šta da kažemo o ovome. Ni malo nam se ne dopada događaj nakon upotrebe podprograma *COMPRESS*. Sada je pitanje samo kada će to opet da se dogodi negde drugde, sve dok ne postane ozbiljniji problem jer neki važan program neće hteti da radi. Istini za volju, to je jedini problem na koji smo naišli, a što je najnazgodnije i on je sporadičan. Za razliku od modela „*Dell*" 320N starija verzija programa *Windows* je radila kako valja.

Na prenosnicima, jedno od glavnih pitanja je uvek pitanje trajnosti akumulatora. Akumulator ugrađen u ovaj računar ima deklaraciju od 1 800 mAh (Panasonic). Prema deklaraciji u priručniku akumulator bi trebalo da obezbeđuje napajanje za „preko dva sata", mi smo odbili vreme od 1 sat i 35 minuta. Istini za volju, prilikom ponovljenog testa, ovaj put sa prilično prigušenim prikazom, prebacili smo 2 sata za 3 minuta. Ovo ne treba da nas čudi, jer prikaz troši oko 40% od ukupne potrošnje računara, disk oko 30% i sve ostalo oko 30%. Dakle, zatamnite ekran i duže će raditi. Najzad, ako usput prebacite i sistemski takt sa 20 na 10 MHz, računajte na oko 2 sata i 20 minuta vremena, što nije loše, ali nije ni malo uzbudljivo.

NEZNATNO BRŽI

Rezultati merenja pokazuju prednost ove mašine nad modelom 320N firme „Dell" koja joj je najsličnija, pre svega zbog bržeg rada sa memorijom imajući u vidu preplitanje. Razlika u procesorskim testovima od 9.4% možda nije velika, ali je ona u programskim testovima pokazala i daleko veće razlike, i od 15%, a u slučaju poslovne grafike, i celih 32%.

U slučaju mašinskih testova i modela „Dell" 320N kao što smo naveli starija verzija programa *Windows* nije htela da radi, iako se od nje tražila obična VGA rezolucija od 640×468 u 16 boja/nijansi. To nas je onemogućilo da završimo testove na regularan način pa nismo mogli ni da rangiramo rezultata kako treba. Zato smo odlučili da video testove apstrahujemo pa otud na našem grafikonu samo tri umesto četiri kategorije za model „*Dell*" 320N, a sve četiri kategorije za model „Chicony".

Na mašinskim testovima „Chicony" je pokazao da se sasvim ravnopravno nosi sa propisnim stonim mašinama u ovoj klasi, istina, onima od juče, jer su te iste stone mašine napredovale do 25 MHz ili 25% veće brzine. No, u kontekstu prenosive mašine postignuti rezultati su čak vrlo dobri: 16,2% bolji od „Dell" 320N i 49,3% bolji od modela „Daewoo" SX/16N (koji, istina, radi na 16 MHz, a postoji i verzija na 20 MHz).

Na programskim testovima ukupna razlika se smanjila na 6,3% u odnosu na „Dell" i 36.5% u odnosu na „Daewoo", pre svega zbog dosta lošeg rezultata na testu sa bazom podataka, gde je „Chicony" zaradio indeks od jedva 0.91. Ko zna zašto, jer se radi o veoma dobrom disku; tek, prilikom ovih testova, potukao je konkurenciju u tri od pet operacija, ali je osetno podbacio u jednoj, indeksiranju baze. Što reče *Dejan* Ristanović (inače i autor tog bloka testova), to je ona operacija koju radite dve ili tri puta godišnje, ali pravila su pravila, kako sa svima, tako i sa ovom mašinom bez obzira što u stvarnosti (odnosno svakodnevnom životu) stvari stoje drugačije nego što bi se iz testa zaključilo. Sve u svemu, brži je od konkurencije, ukupno za skromnih 6,3%.

DOBRO I LOŠE

U principu, ovaj računar nam se veoma dopada, a otvoreno priznajemo da smo skloni firmi „Chicony", najzad, već godinama vucaramo unaokolo prenosnik LT 3600. Veoma nam se dopadaju neke osobine ovog računara, a sa nekima nismo oduševljeni (recimo načinom proširivanja memorije i nedostatkom provere pariteta). Ekran je natprosečno dobar.

Kao vlasnici starijeg modela istog proizvođača, mi bismo rado prešli na noviji, jer je objektivno praktičniji. Između modela „*Dell*", „Daewoo" i „Chicony", verovatno bismo se opredelili za „Chicony" iz niza razloga uz ogradu da bismo ipak prvo probali slično opremljen model „Daewoo" (to jest sa isto tako 2 MB memorije i na taktu od 20 MHz).

Da li ćete i vi preći na njega, verovatno će zavisiti od vaših potreba i cene. Ako ona bude konkurentna, onda svakako pogledajte ovaj model.

Korisna adresa

Osa
Miće Stojkovića 8, 11130 Beograd
Tel. 011/557-592, fax 011/559-645

ŠTA IMA NOVO

DELTAGRAPH PRO ZA WINDOWS

Firma „DeltaPoint" je najavila paket za izradu grafikona i prezentacija pod nazivom *Delta Graph Professional*, veoma nalik istoimenoj „Macintosh" aplikaciji istog imena. Kompatibilna sa *Windows*-om *3.1*, nova aplikacija prepoznaje OLE i podržava TrueType fontove. Jedna od ključnih opcija novog paketa je *Chart Advisor*, opcioni alat za izbor odgovarajućeg grafikona koji raspolaže sa 40 tipova dvo- i trodimenzionalnih grafikona, automatskim osvežavanjem grafova povezanih sa *Excel*-ovim podacima i mogućnošću uvoza i izvoza grafike u velikom broju formata. Pored toga, uz program se isporučuje i biblioteka od preko 500 *Clip art* sličica. *DeltaGraph Professional for Windows* je uz to sposoban za automatsko prikazivanje „slajdova" i štampanje ekrana. Novi paket je, prema rečima predstavnika firme, namenjen kako naučnom tako i poslovnom krugu korisnika po ceni od 495 USD, uz početak prodaje krajem drugog kvartala.

Kontakt adresa: DeltaPoint Inc. 2 Harris Court Suite B-1, Monterey, CA 93940 (408) 648-4000. Fax, (408) 648-4020

ROOMS FOR WINDOWS

Rooms for Windows je paket firme „XSoft", jedne u lancu „Xeroxa", čija namena je povećanje virtualne veličine desktopa u *Windows* okruženju. Ovaj paket dozvoljava kreiranje više različitih desktopa koji su predstavljeni odgovarajućim ikonama, a pozivaju se klikom miša. „Xerox" je patentirao glavnu komponentu paketa – mogućnost učitavanja iste aplikacije na više lokacija istovremeno. Korišćenje *Rooms for Windows* bi se moglo predstaviti kao korišćenje nekoliko radnih stolova u istoj kancelariji, pri čemu se na svakom stolu nalazi različit asortiman projekata u radu (otvorenih fajlova), tako da je korisnik u stanju da lako prelazi sa jednog stola na drugi. Ali, za razliku od pravih stolova, *Rooms* je u stanju da istu grupu programa i fajlova drži na više desktopa istovremeno, ili čak iste DOS programe sa različitim učitanim fajlovima. Sam *Windows* operativni sistem ne dozvoljava višestruko otvaranje iste DOS aplikacije.

Za razliku od *Norton Desktop*-a, *Rooms* ne zamenjuje *Program Manager* kao radnu sredinu, već samo prihvata *Windows* aplikacije, pri čemu koristi sve prednosti 386 virtualne memorije za preklapanje desktopa. Takozvana *Overlay Room* predstavlja neku vrstu maske koja definiše stavke koje treba da se pojave u svakoj pojedinačnoj „sobi" (*Room* – prostor, soba). Otvoreni fajlovi i korisnički definisana dugmad (koja se mogu koristiti za učitavanje fajla i aplikacije istovremeno) mogu se kopirati između pojedinih „soba". Broj soba i otvorenih aplikacija zavisi od sistema: pod *Windows*-om *3.0*, prema „Xsoft"-u, *Rooms* je u stanju da istovremeno otvori 11 aplikacija, dok pod verzijom *3.1* ovaj broj raste na 30, a u laboratorijskim testovima je postignuto 70 „soba" istovremeno.

Jedini problemi koji su se do sada pojavili su nastali pri radu programa koji na neki način krše pravila *Windows*-a. Na primer, *Norton Desktop* crta neke od svojih ikona direktno na ekranu, tako da ih *Windows* zapravo i ne registruje. Budući da *Rooms* koristi standardne *Windows* API pozive za kontrolu, premeštanje i promenu dimenzija prozora, nije u stanju da kontroliše ove *Norton*-ove ikone. Drugi potencijalni problem se javlja sa konkurentskim paketima koji takođe koriste metaforu „soba".

NOVI MT LASER

Novi modeli laserskih štampača se ne izbacuju na tržište ni izbliza onom brzinom kojom se pojavljuju računari, pa čak i aami procesori. Zato je uvek novost kada se pojavi sasvim novi model, u očekivanju da će biti bolji od prethodnih u sopatvenoj porodici i od – konkurencije.

"Mannesmann-Tally" je ogranek firme „Mannesmann", koje se bavi investicijama. Odnedavno, od ove firme je ostalo mahom ime, jer se deo sjedinio sa „Siemensom", a deo sa „Kienzleom" koga je, opet, kupila američka firma DEC. Uprkos svemu ime je ostalo iako je danas očigledno svedeno samo na tržišni znak. U osnovi firme je locirana u Nemačkoj, ali već poodavno se proizvodi preve u Japanu – istina, po nacrtima i specifikacijama Nemaca. Naši verniji čitaoci će se setiti da smo o njihovom modelu MT 222 imali „superlativno" mišljenje (i dan-danas ga smatramo najboljim štampačem tog tipa koji se ikada pojavio kod nas), a dobro smo mislili i o starijem modelu lasera MT905 i o njegovom nasledniku MT906.

U to vreme MT je nudio i veći i oho-ho jači model 910 koji je sem bio dve puta brži, 'em je podržavao *PostScript*; ukratko, bio je namenjen daleko profesionalnijem segmentu tržišta. Model MT905 je u osnovi bio TEC mehanike i elektronike pod 1001 imenenom – „Epson" 6000, „Facit" 6060, MT 905 i još tridesetak imena za koja mi znamo. Ukratko, bio je to tipičan primer generičkog proizvoda: pravi ga jedan, a prodaje ih sto. Iako se radi o zaista dobrom proizvodu, taj štampač je imao i nekoliko mana, pre svega doste rasipničko ponašanje sa memorijom, zbog čega smo dolazili u situaciju da on sa 1,5 MB ne može ono što HP može sa 1 MB. A naravno, tu je bila i brzina rada, procesor „Motorola" 68000 na 10 MHz, srce te serije, više nikoga ne može da uzbudi.

Model MT906 je bio veliki korak napred: imao je drugačiji procesor, „National Semiconductor" 32016 (32 bita iznutra, 16 bita spolja) umesto M68000, radni takt od 15 umesto 10 MHz i potpuno drugačiju elektroniku sa istom mehanikom. Sve zajedno oko dve puta brže za rad i daleko elegantnije urađeno od modela 905.

Dejan V. Veselinović

jer tako koncipirane mašine uvek imaju veoma skupa memorijska proširenje. Jedino što vam može ići u račun jeste da i ovo postane široko prihvaćen generički model, u kom slučaju će se, kao i za stariji model 905, pojaviti jeftina proširenja iz trećih izvora (za oko 1/4 cene originalnih MT delova).

Malo smo začuđeni da „Ibis Sys", firma koja svoje ime u velikoj meri duguje baš MT štampačima, uopšte nudi ovaj štampač u verziji sa 512 KB, čime dolazi do inverzije situacije – umesto da 1,5 MB bude standard pa da se posebno pregovara za manje (ako nekome baš to treba), ovako morate da razgovarate za ono što vam danas po definiciji treba. Podsećamo da A4 stranica sa grafikom zahteva najmanje 1 MB memorije. Što je još interesantnije, rečeno nam je da prodavac trenutno uopšte na raspolaže memorijskim proširenjima (?!), što je u najmanju ruku čudno za bilo koga, a posebno za kuću sa ovakvim renomeom u prodaji štampača (istini za volju, proširenja su naručena i na putu su).

Dakle, proizvođač same elektronike je „Konica", a proizvođač mehanike je opet japanska firma „Matsushita" („Technics"/"Panasonic"). Pretpostavljamo da je ovo bio nužni korak u kontekstu želje da se ubrza štampanje, jer je TEC-ova mašina bila nepodobna za ubrzavanja, što je bila pretpostavka novog modela.

Sve ostalo je prilično standardno. Dimenzije štampača su 450×415×235 mm (ŠxDxV), težina solidnih 19 kg, a kaseta sa papirom povećava dubinu za nekih 15 cm, što je bolje od proseka, ali gore od recimo modela HP IIIP čije dimenzije rastu u visinu, a ne u polje (dakle, takvo rešenje je neuporedivo bolje). Potrošnja energije je oko 800 VA kada radi, odnosno oko 130 VA u stanju pripravnosti.

JAPANCI ZA NEMCE

A sada dolazi model MT911. On je, po logici stvari, dalja razrada modela 906. I zaista kada ga otvorite, elektronika iznutra veoma nalikuje modelu 906. Tu je procesor NS 32016 opet na 15 MHz, VLSI kolo sa oznakom VGT8005-6177, mesto za osam ROM čipove od kojih su popunjena četiri, 4 RAM čipa (4256, ukupno 512 KB), postolje za još toliko (uzgred, ta postolje se ne pominju u priručniku, pa se može desiti da su ona samo alternativa ZIP modulima koji su ulemljeni), mesto za dva font proširenja (u vidu nestandardnih odnosno HP nekompatibilnih kaseta) i naravno mesto za proširenje ukupne memorije.

Sve to je uradno potpisala japanska firma „Konica", za koju smo do sada verovali da proizvodi samo foto-aparate. Ako je suditi po ovoj ploči, prvi koraci u elektronici su im veoma dobri.

Od memorijskih proširenja, možete kupiti tri pločice, sa 1, 2 ili 4 MB. Pošto je malo verovatno da će ovaj štampač ikada dostići nivo popularnosti uzora, modela HP, verovatno je da ćete biti pupčanom vrpcom vezani za proizvođača. Ovo je uvek mana, bar u osnovnom smislu,

MEHANIKA

U poređenju sa starijim modelima, napravljen je zaista veoma veliki korak napred. Mehanika je urednija, odnosno bolje i funkcionalnije raspoređena, ali nije žrtvovano bilo šta od kvaliteta izrade. Štaviše, pre bi se moglo govoriti o poboljšanju, primera radi, umesto jednog sada se u štampaču nalaze dva rashladna ventilatora, od kojih jedan stvara potpritisak a drugi izbacuje vruć vazduh van kutije.

Mehanika je zapravo podeljena na dva dela. Jedan se nalazi čvrsto montiran unutar osnove kutije i sadrži delove koji se veoma retko menjaju, a drugi deo montiran na poklopac koji se otvara na šarkama i sadrži delove koji se češće menjaju. Mada i to „češće" treba malo razjasniti – recimo, selenski valjak ima životni vek tačno duplo duži od prethodnika (20 000 umesto 10 000 stranica), a kaseta sa tonerom, večni problem i mora vlasnika lasera, sada traje za oko 2 000 umesto 1 500 stranica. I dalje nam nije jasno zašto to nije moglo i celih 3 000 stranica, mesta ima i za više, ali sad, šta je, tu je.

Pošto se doboš i toner menjaju odvojeno, primenjeno je do sada najelegantnije rešenje koje smo ikada videli – toner je pristupačan sa spoljne strane kutije, a nalazi se ispod jednog posebnog poklopca. Možete ga zameniti za manje od minute, apsolutno bezbedno i bez prosipanja praha, pa makar i da nikada niste ovako nešto ranije radili! Ovo je toliko lepo rešeno da nam samo ostaje da se nadamo da će i drugi prihvatiti ovakvo rešenje. Ako je HP elegantan, onda je ovo poslednji krik mode.

ELEKTRONIKA

Štampač emulira dobro poznati model HP II, ali na njega dodaje niz sopstvenih komandi. Nema tu bilo čega radikalno novog, sve je već viđeno kod ranijih modela. Po želji i uz doplatu,





emulirače i „Epson" FX80+ i IBM „Proprinter" XL, a postoji i *PostScript* kaseta.

Jedino po čemu se vidi da je sve to novijeg datuma je broj fontova koji se standardno isporučuju, pored obaveznog **Courier**-a (10 i 12 punkta, masno i kurziv), tu su i **Dutch/Times Roman** (6, 10, 12 i 14 punkta, masno i kurziv, proporcionalno) i **Swiss/Helvetica** (6, 8, 10, 12, i 14 punkta, masno i kurziv, proporcionalno). Neki od tih fontova postoje i u „portrait" i u „landscape" verziji, efekat koga na starijim modelima nije bilo.

Od nizova znakova, tu su svi poznatiji i priznatiji evropski i azijski jezici (nemački, španski, portugalski, norveški, francuski, švedski i kineski). O nama, naravno, nema ni reči.

Meniji pomoću kojih se postiže željeno podešavanje nisu ni bolji ni gori od konkurencije, niti od prethodnih modela. Ima ih logičnih, a ima ih i zakukuljenih do zla boga, pa morate do njih zaista okolo-naokolo (ne momente, žarko smo poželeli da se vratimo u doba DIP preklopnika). Jedna stvar koja nam se dopala jeste da sa menija **FNT Enhance** možete podesiti da svi fontovi, računajući i one koje „spuštate" sa računara, rade u oba smera, i po vertikali i po horizontali.

MERENJA

Od deklarisanih 10 stranica u minutu, ne muku smo uspeli da isteramo 8,2 stranice u minutu, ovo je dobar rezultat kada se ima u vidu da je model HP IIP deklarisan za 8 a daje 4,1 stranicu u minutu, a model III deklarisan za 8 a daje 6,3 stranice u minutu. Izgleda da je brzina mamac kome ni jedan proizvođač štampača ne može da odoli, pa su deklaracije uvek naduvane. Ovo u stvari uopšte nije loš rezultat, po deklaraciji, MT911 je samo 25% brži od HP II/III, a u stvarnosti je 30,1% brži. Preteruju i jedni i drugi – ali šta je, tu je.

Sliku sa senčenjem MT911 je završio za 22,51 sekundu, poređenja radi, HP III to isto uradi za 33,41 sekundu. Demo sliku sa kućom i akvaduktom (iz programa *PC Paintbrush*) 911 završi za 31,66 sekundi, a HP III za 47,17 sekundi. Drugim rečima, u pukoj praksi, ovaj štampač je veoma osetno brži od HP III. Osnovni razlog ovoj brzini nije toliko mehanika koliko procesor i radni takt, oba su kod modela 911 bolja ili brža od HP III.

BRZO I... BUČNO

Na žalost, nemamo instrument za merenje buke, ali kao korisnici, ne bismo rekli da je ovo baš mnogo tih štampač. Dok je u stanju mirovanja zaista se živ ne čuje, ali kada počne da radi, odnosno kada počne da uvlači papir, javlja se jedan ne mnogo prijatan zvuk. Nije preglasan, ali nekako deluje kao da se štampač muči. U praksi, ne bi se moglo reći da naročito pati, jer ipak izbaci solidan broj strana, ali zvuk je obeshrabrujući kada ga prvi put čujete (posle se naravno naviknete a sve lepo radi). Istini za volju, uzorak sa kojim smo mi radili je bio potpuno nov, mi smo ga prvi izvukli iz originalnog pakovanja, pa po logici treba očekivati da će nakon razrade i pomenuti zvuk ili nestati ili se manjiti po glasnosti.

Subjektivno imate osećaj da je MT911 mnogo brži nego što stvarno jeste. Recimo, kada „spuštate" fontove iz računara u njega, ceo proces koji na starom modelu 905 traje 2-3 minuta ovde se završi za 35-50 sekundi. Dalje, u celini gledano, čak i mehanički ovaj štampač je duplo brži, što povećava subjektivni osećaj brzine prilikom štampanja složenijih i dužih tekstova. Vrhunac osećaja brzine dolazi kada štampate duže datoteke sa dosta grafike, imate osećaj da prosto leti.

Najzad, cela mašina deluje dosta robusno i nekako uliva poverenje, sa izuzetkom onog čudnog zvuka prilikom uvlačenja papira. Sa tim u vezi, još jedna opaska – štampač je dosta osetljiv na ispravno umetnut papir. Ne sme da bude izdignut ni za milimetar, pa nemojte da vas mrzi da lepo izvadite kasetu i pažljivo ubacite papir, svih 200 listova (75 gramskog papira). Ukoliko ne postupite tako, može se dogoditi da dođe do zaglavljivanja papira. Iako je ovo bez sumnje neprijatno, nije i tragično, jer kada štampač otvorite, cela putanja papira se nalazi odmah ispred vas. Inače, senzor zaglavljivanja je prilično osetljiv, odmah stupa u akciju, pa nema mnogo prilike da se papir iscepa na sitne komadiće koje posle ne možete ni pronaći, a kamo li izvaditi. To mu dođe nešto kao sreća u nesreći.

Teško je ne zaključiti da je ovo dobar štampač, sa brzinom koja je iznad proseka i elektronikom koja je među najboljim u klasi standardnijih štampača. Međutim, sem toga, on ne nudi bilo šta drugo što bi ga posebno preporučivalo – niti ima neke posebne osobine, niti nudi neko obilje fontova, niti koristi standardne HP kasete, niti koristi standardne delove za memorijsko proširenje, niti ima onako veliku prodajnu i servisnu mrežu kao HP.

Pogodan je za kancelarije i to upravo zbog brzine rada, kao i za one kojima treba veliki broj stranica na dan, pa razlika od 30% u brzini igra prilično važnu ulogu. Prednost je i to što doboš (skup deo) traje 20 000 stranica i što se toner menja kao poseban deo, to znači da je ovaj štampač ekonomičniji za održavanje od konkurencije koje koristi mehaniku *Canon*-a.

Ukoliko prodavac, firma „Ibis Sys", uspe da obezbedi solidnu cenu (provenit pre kupovine usled deviznih kolebanja) i rezervne delove, i ako vam treba brzina posebno u kancelarijskim uslovima, svakako ga uzmite u obzir. Ne zaboravite da pitate koliko košta sa dopunskih 1 MB memorije i da li ih uopšte imaju. Kupujući ga, uživaćete u jednom zaista solidno napravljenom proizvodu, robusnom sa jedne i prilično rafiniranom sa druge strane.

U svim drugim slučajevima, kupite HP IIIP i on radi na 15 MHz, koristi standardne delove i proširenja, ima varijabilnu veličinu tačke i ima ga gde hoćete, sa servisom. I jeftiniji je.

Korisna adresa

Ibis Sys
Panske komune 22, 11070 N. Beograd
Tel. 011-609-650, Fax 011-699-081

ŠTA IMA NOVO

Nagrade

PRIZNANJE „CORELA" STIGLO U SARAJEVO



Sa izvesnim zakašnjenjem smo saznali za jedno priznanje koje je nedavno stiglo na eks-jugoslovenske prostore. Na redovnom mesečnom konkursu „Corel Draw World Design Contest" za februar ove godine **Nebojša Blagojević** iz sarajevske firme „Guard Design" dobio je drugu nagradu u kategoriji tehničkih crteža. Nagrada podrazumeva i svetsku prezentaciju i objavljivanje u publikaciji „Corel Creative Club".

Korporacija „Corel" proizvodi zvanično najbolji program za grafički dizajn i drži 77 odsto svetskog tržišta u ovoj oblasti, sa preko 200 000 registrovanih korisnika. Mesečni konkursi su deo Svetskog dizajn takmičenja (World Design Contest) vrednog milion dolara! Crtež Nebojše Blagojevića (pištolj „Ruger P85") bio je jedini sa eks-jugoslovenskih prostore. Urađen je u paketu *Corel Draw 2.01* bez drugih pomoćnih alata i sadrži preko 2 200 objekata. Dobitniku druge nagrade upućen je poziv da na takmičenje pošalje nove crteže, čiji broj nije ograničan, kako bi konkurisao za glavne nagrade.

Operativni sistemi

OS/2 2.0 NAJZAD U PRODAJI

IBM je konačno uspeo. Razvoj komercijalne verzije 32-bitnog operativnog sistema koji u jednom paketu objedinjava sve ono što nude DOS, *Windows* i OS/2, sa oznakom OS/2 2.0 je konačno pušten u serijsku proizvodnju. OS/2 2.0 sadrži grafički korisnički interfejs pod nazivom *Workplace Shell* i omogućava rad OS/2, DOS i *Windows 3.0* aplikacija. Podržane su VGA, XGA i 8514/A video kartice pri radu sa *Presentation Manager*-om i *Windows* aplikacijama, dok SuperVGA mod još nije podržan „ali se na tome radi". Ovo je naišlo na oštre kritike korisnika i proizvođača softvera, ali to ipak nije zadržalo početak prodaje novog paketa. IBM tvrdi da pod OS/2 2.0 radi više od 17 000 DOS, 4 900 *Windows* i 2500 16-bitnih OS/2 aplikacija, a da preko 1 000 proizvođača trenutno razvija 32-bitne aplikacije za OS/2 koje će se naći u prodaji tokom godine, pomenuta su i imena kao što su „Lotus", „Borland", „WordPerfect" i „Novell". Za sada je preko 100 modela računara prošlo testove kompatibilnosti uključujući „Compaq"-ove, „Dell"-ove, „CompuAdd", AST i „Tandy"-jeve modele, a posebna pažnja je posvećena 386 i 486 platformama.

OS/2 2.0 se isporučuje po preporučenoj ceni od $139, registrovani korisnici *Windows* paketa treba da plate $49, korisnici DOS-a $99, dok korisnici 16-bitne verzije OS/2 dobijaju besplatni *upgrade* do 31. jula ove godine ako ga naruče na (američki) telefon 800-3-IBM-OS2. Na žalost, ovaj broj se može zvati samo iz Amerike.

DVA U JEDNOM

Kada smo u januaru prvi put objavili tekst o faks/modem kartici, desilo se nešto neočekivano. Redakcijski telefon je nekoliko dana bio okupiran pozivima čitalaca koji su tražili (tehničkom omaškom) izostavljenu adresu. Zbog toga smo odlučili da toj vrsti proizvoda posvetimo još malo pažnje. Opredelili smo se za dva faks/modema, međusobno potpuno različita u svemu osim nameni, koja možete kupiti odmah, ovde kod nas.

Da se podsetimo – faks/modem uređaji su kombinacija klasičnog modema sa telefaks ćeljkom. U ne tako davno vreme konstili su se telefaksi koji su mogli samo da šalju ali ne i da primaju poruke. To je, međutim, prošlost – danas se podrazumeva pun skup osobina koje se očekuju kod bilo koje pristojne faks mašine, pre svega mogućnost slanja i prijema. Danas se već očekuje da rad sa telefaksom bude na nivou standarda, odnosno trenutno važeće Grupe 3 (koja obuhvata prethodne grupe 1 i 2). Što zapravo znači da bude sposobna da uspostavi kvalitetnu vezu sa ogromnom većinom takvih mašina u svetu, čiji se broj procenjuje na oko 70 miliona (prognoze navode broj od preko 200 miliona do 1995. godine).

Faks na kartici ima nekoliko značajnih prednosti i mana u odnosu na stone telefaks mašine. Od prednosti treba pomenuti daleko skromniju cenu, iste ili nešto veće mogućnosti slanja i prijema i nedostatak potrebe za posebnim prostorom i potrošnim materijalom, a od mana ostaju faktički samo dve: ako vam treba unos neke slike, moraćete to da obavite posebnim skenerom, a kada ugasite računar – nemate više faks.

Ova poslednja mana nije nužna, recimo, već pre dva ili tri godine, neki modeli američke firme „Brooktrout" su mogli da razaznaju vrstu poruke, uključe računar, prime faks poruku i potom ponovo isključe računar. Ovo je bilo moguće tako što se kartica napajala iz autonomnog izvora, a preklop za uključenje se vodio preko nje same. Sasvim očekivano, problem je bila cena od nekih 600 USD, što je i u to vreme bilo podosta.

Danas je osnovni adut ovakvih kartica upravo cena, koja je za mnoge odlučujući faktor, a posebno za povremene korisnike faks mašine. Zašto biste plaćali dva i po ili tri puta više za stonu faks mašinu ako sve to može jednostavnije i lakše ovako? Na 10-20 faks poruke nedeljno, ovakve kartice su zaista bolje rešenje.

Ova problematika se posebno zaoštrava u slučaju vlasnika i korisnika prenosivih računara. Ideja je da oni sa sobom, umesto samo računara, nose pravu komunikacionu stanicu, a faks/modem uređaji im upravo to i omogućavaju. Dok su prenosivi računari bili poveći, kao recimo naš stari model „Chicony" LT 3600, nije bio poseban problem ugraditi standardnu karticu u standardnu utičnicu za proširenja. Sada kada se prave mašine formata A4 ili manje, ovakav pristup postaje nemoguć. Zato, ili ćete kupiti takvu karticu koju nudi sam proizvođač računara ili ćete se opredeliti za spoljni uređaj. Taj uređaj opet mora biti primeren računaru: a ne neka velika metalna kutija, teška i kabasta kao i sam računar. Najzad, za takav stepen minijaturizacije, očekujte da ćete platiti najmanje 30-50% više, a katkade i 2 ili 3 puta više nego za prostu karticu namenjenu standardnim AT mašinama.

Da bismo sve ovo proverili i u praksi, ovaj put smo probali za vas dva sasvim različita uređaja sa istom namenom.

Dejan V. Veselinović

DYNALINK 9624P

Odmah da kažemo – ovaj uređaj je prirodni partner prenosive mašine „Chicony" NB5620 koju proizvodi isto preduzeće, od koncepcije (putuj lagano ali totalno opremljen) do boje (ista tamnosiva boja). Radi se o uređaju koji je veličine otprilike jedne i po paklice cigareta. Smešten je u tamno sivu plastiku koja se završava jednom DB-9 „ženskom" utičnicom, namenjenom sličnom RS 232 C serijskom vezniku na matičnom računaru.

Na kutijici se nalazi preklop za uključenje i isključenje, rotacioni zupčanik za pričvršćivanje za računar, jedna ulazna i jedna izlazna utičnica (LINE i PHONE) i najzad mala rupica za dovod napajanja. Ovo je inače opciono, jer unutar kutijice možete staviti dve IEC 6 tranzistorske baterije od po 1,5 V, čime postajete zaista veoma pokretljivi.

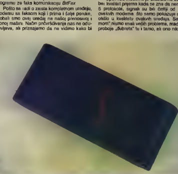
U kutiji ćete takođe naći i spoljni izvor napajanja, jedan kabl za povezivanje sa telefonom (AT&T-Bell, odnosno američka utičnica), tri priručnika i dve diskete od 3,5 inča. Jedan priručnik obuhvata sam uređaj (instalacija, AT komande itd), drugi se odnosi na program za modemsku komunikaciju *BitCom*, a treći služi programu za faks komunikaciju *BitFax*.

Pošto se radi o zaista kompletnom uređaju, modemu sa faksom koji i prima i šalje poruke, probali smo ovaj uređaj na našoj prenosivoj i stonoj mašini. Način pričvršćivanja nas ne oduševljava, ali priznajemo da ne vidimo kako bi to moglo bolje, zupčanici od plastike nam baš ne ulivaju poverenje.

Takođe u vezi sa montiranjem, kada pričvrstite uređaj, odmah dodajte delijh dvadesetak centimetara potrebnog prostora do zida. Nije baš mnogo praktično, ali na žalost jeste nužno. Nezgodno je i to što većina prenosnika ima sve veznike sa zadnje strane, koji se kada sedite ispred ekrana ne vide, pa samim tim ne možete videti ni signalne diode na uređaju koje vas obaveštavaju o trenutnom stanju rada. Najzad, ni malo nam se ne dopada veoma bedunjev ispis šta je šta i šta je gde na samom uređaju, na kome se normalno vidi samo naziv modela. Zar je baš moralo tako anemično?

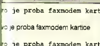
Moramo priznati, međutim, da smo uređaj bez po muke montirali na obe mašine. Kao i svi dobri uređaji ovog tipa, tako i ovaj nudi izbor između impulsnog i tonskog biranja, međumemoriju za komande od 40 znakova (prosečna vrednost), a konfiguracija se čuva u neizbrisovoj memoriji (Low Power CMOS). Tolerancije radnih učestanosti su prema deklaraciji veoma dobre, čak ± 0,01% (bolje od našeg MNP 5 internog modema). Čak je ugrađena i neka vrsta kliznog prilagođavanja impendanse, opet ne baš svakodnevna pojava (mada noviji modemi to imaju). Na žalost, uređaj ne podržava protokol MNP 5 u hardveru.

Upotreba modema je ukazala na zaista dobar kvalitet prijema kada se zna da nema MNP 5 protokola, signali su bili čistiji od proseka ovakvih modema, što samo pokazuje dokle se otišlo u kvalitetu ovakvih uređaja. Sa „Sezamom" nismo imali većih problema, mada je bilo proboja „đubreta" tu i tamo, ali ono nikada nije



Snimio: Nenad Marijanović

Fax/modem za prenosne računare. Dynalink 9624P



Bolje od klasične faks mašine. Zahvaljujući prirodnim prednostima računara, faks kartice prenose tekst veoma kvalitetno

trajalo dugo i ni jednom nas nije nateralo na Konačno Rešenje (Alt-H). Sa ETF-om, zbog poznatih problema sa nekvalitetnim linijama (u pristojnom društvu, to se tako zove, inače, smatramo da bi PTT zaista trebalo da prekine sa upotrebom kanapčića umesto bakarne žice), nije moglo da se uradi mnogo bez spoljnih programa za emulaciju protokola MNP. Najzad, prilikom uspostavljanja računar-računar veze sa kolegom na Novom Beogradu, nismo imali bilo kakvih problema sa slanjem i prijemom datoteka.

Dakle, modemski deo se može smatrati dobrim do vrlo dobrim. Deo za faks poruke je u celini ispunio naša očekivanja, jednostavno tako što je radio bez ikakvih problema. Malo smo se mučili sa *BitFax*-om, ali kada smo ga naterali da radi, sve je bilo u redu. Kao i svi ovakvi faks uređaji, tako nas je i ovaj oduševio rezultirajućim porukama kod pošiljaoca, ovo je, naravno, normalna stvar, jer ovakvi uređaji zaobilaze proces skeniranja sa svega 200×200 tačaka (po standardu, i do 300×300 tačaka na po nekom modelu, ali to nije standard) po kvadratnom inču, pa je tekst poruke izvanrednog kvaliteta (vidi primere). Ni sa prijemom poruka nismo imali problema, sve je radilo kako treba.

Konačno poređenje ovog uređaja nužno smo izveli prema našem ATI 2400etc modemu (2400 bps, MNP 5 u hardveru) i „Canon" FAX-80 uređaju. Naš modem je naravno bolji za onaj MNP protokol, a u slučaju faks mašine zaista ne vidimo ni jednu jedinu prednost stonog modela, sem onih koje mu po prirodi stvari pripadaju (uključite ga i on ako treba stalno prima, mogućnost kopiranja papira na licu mesta i slično). Programi koji se dobijaju uz uređaj se moraju smatrati sasvim zadovoljavajućim, *BitFax* ionako standardno koristimo za naš interni faks/modem uređaj (jer podržava prenos iz raznih grafičkih formata i raznih programa za obradu teksta), ali vam *BitCom* ne bismo preporučili kao komunikacioni program – 'em se radi o starijoj verziji, 'em po nama nije ni malo udoban za upotrebu, uprkos sasvim pristojnim opcijama. Njega odmah zamenite nečim boljim, naš savet vam je „Procomm" 2.1.

Minijaturizacija ima svoju cenu, u ovom slučaju oko DEM 900 kod „Osa" ili nekog od njihovih autorizovanih dilera. To je zaista dosta, ali ipak ne možemo reći da uređaj ne vredi te pare, uz ogradu „ako vam treba za prenosni računar". U tom slučaju, nema dileme da li da ga uzmete ili ne, kupite ga. Međutim, ako vas zanima ovakav uređaj za neku stonu mašinu, pogledajte druge modele kod drugih ponuđača, koštaće vas podosta manje.

Najzad, „Osi" sledi pohvala za ponudu ove nadogradnje na prenosnike, odnosno promišljeno zaokruženje svoje (a i tuđe) ponude u ovom segmentu, dobar izbor, uprkos ceni.

RIPEFAX/MODEM

Nakon prve Tseng ET4000 kartice sa 33 000 boja, „Mikronova" je, izgleda, odlučila da nastavi da nudi lepe proizvode na našem tržištu. Ovaj put su nam, u okviru svoje standardne ponude, na testiranje ponudili faks/modem karticu koju nude pod umetničkim imenom „RipeFAX/modem".

Naravno, radi se o faks/modem kartici koja je namenjena ugradnji u sasvim standardne računare, od modela PC/XT do AT/486. Sa izuzetkom četiri induktora, koji su naherani na sve strane sveta, kartica je dosta uredno napravljena na bazi već dobro isprobane tehnologije. EXAR 2901 i 2902 čipa, u sprezi sa jednim „Intel" 8031 komunikacionim procesorom. Svi glavni čipovi se nalaze na postoljima, čime se dobija visok stepen servisabilnosti, uz na žalost i nešto veću cenu. Jedina manja zamerka proizvođaču je to što su DIP preklopnici za podešavanje COM veze montirani sa unutrašnje strane uređaja i nisu pristupačni spolja, kažemo manja zamerka, jer objektivno nema veće potrebe svaki čas menjati COM vezu, posebno kada se zna da odgovarajuće izmene treba obaviti i u podešavanju programa. Ipak, voleli bismo da je to spolja moguće uraditi.

Karticu smo bez ikakvih problema ugradili u naš računar (prethodno smo izvadili naš modem, zlu ne trebalo). Uz put smo nekoliko puta izgrdili proizvođača zbog ne baš sjajnog programa samo za faks komunikaciju, a onda smo seli da bar njega instaliramo. I gle čuda, umesto Hromog Dabe, dobismo „Dete Grujicu"! Prvo, program se isporučuje na dve AT diskete, a obuhvata celokupno komunikaciono rešenje – drugim rečima, modemske komunikacije se nalaze u okviru celine. I ne samo to – prilikom izbora nacionalnog niza znakova, pod brojem 15 ugledasmo „Yugoslavia" (i to pre kanadskih Francuza).

Prvo smo probali modemski deo. Konstili smo uputstva iz priručnika, koji ujedno opisuje elektroniku, i oba dela komunikacionog programa. Modemski deo smatramo vrlo dobrim, marginalno boljim od onog u „Dynalink" uređaju (mada odmah napominjemo da se radi o subjektivnom utisku). Sa „Sezamom" ni u ovom slučaju nismo imali problema, a sa ETF-om smo nešto manje puta morali nanovo da uspostavljamo vezu (što može biti i usled nešto boljeg stanja linije). Ipak, nekako nam se čini da je ovo najbolji klasični modem sa kojim smo imali posla do sade, mada nam je veoma žao što nema ugrađen MNP protokol (prostora ima, bio bi nešto skuplji, ali i osetno bolji).

U slučaju faks odeljka, važi sve što smo naveli i za model „Dynalink". Krajnji rezultati su identični po izgledu. Najveća razlika između ova dva modela se zapravo odnosi na programe koje uz njih dobijate. Pri tom, dajemo jasnu prednost ovom modelu odnosno programu („RFAX", autori Jang & Jong) koji je jednostavniji od „BitFax"-a, ali bez ikakvih gubitaka na efektima. Poruka kod primaoca je potpuno istovetno kvalitetna, a rad je nešto lakši, odnosno komforniji. Iako se radi o malo poznatom programu, spisak podržanih opcija je vrlo dobar, a spisak podržanih grafičkih formata čak odličan (na ulazu: TIF, PCX, PCC, CUT, MSP, CAP, FAX, MAC i IMG, a na izlazu nakon konverzije po želji: PCX, FAX, TIF i IMG). Da smo na mestu autora, prodavali bismo ovaj program i sam za sebe, iako nije ni izbliza onako ispoliran i našminkan kao recimo *BitFax*. Prava šteta što bar neka verzija nije u javnom vlasništvu – verujemo da bi stekao svoj krug vernih korisnika.

Jedina pomalo čudna stvar se nalazi na meniju za definisanje skenera koji koristite: na kome nema ni pomena od bilo kog HP uređaja, a navodi se niz ručnih skenera tajvanske proizvodnje (računarski nacionalizam?).

Sa prodajnom cenom od oko DEM 600, ovaj uređaj je za stone mašine svakako bolji posao od „Dynalink"-a. To je oko 40% prodajne cene stonih faks mašina, ali naravno uz sve pomenuta ograničenja i bez sekretarice i telefona. Ipak, za povremene korisnike faks mašina, kao i za one kojima treba da pojačaju postojeću komunikacionu opremu, ovo se mora smatrati veoma dobrim do odličnim poslom (da modem ima MNP5 u hardveru, bio bi to čist odličan posao). Glavna vrlina – zreo proizvod, bez dečijih bolesti, po dobroj ceni.

Dakle, „Dynalink" je tehnički veoma usavršen proizvod, namenjen jednom jasno određenom segmentu tržišta, čije potrebe sasvim dobro zadovoljava. „RipeFAX/modem" je opet okrenut najširim masama i stonim mašinama. „Dynalink" sve karte baca na svoju specifičnost u smislu oblika i namene, dok „RipeFAX/modem" igra na celovitost paketa i zrelost tehnologije. Tehnički, divimo se „Dynalink"-u, pa u tom kontekstu smatramo da nije preskup, ali ako moramo da biramo po celovitosti, onda bismo odmah odabrali „RipeFAX/modem". On ne impresionira po tehnici, već po veoma lakom uključenju uređaja u red i komfornoj upotrebi (a ni niža cena nam ne smeta).

Ako su ova dva proizvoda vesnici stavova njihovih prodavaca, i ako prodavci nastave da razmišljaju i rade na ovaj način, onda nas čekaju lepa (makar računarska) vremena. Oba proizvoda su puni pogodak za tržišne segmente kojima su namenjeni.

Feks/modemi

	MODEM Datalink	RFAX/Mod
Brzina rada	2 400	2 400
Automatsko prilagođavanje	Da	Da
MNP protokol/klasa	–	–
Sopstvena serijska linija	–	Da
Podešavanje COM linije	–	Da
Standardi		
Bell 103/212A	Da	Da
CCIT V 21	Da	Da
CCIT V 22	Da	Da
CCIT V 22 bis	Da	Da
Priloženi program(i)	BitCom	RFAX
FAKS		
Samo slanje	–	–
Prijem/slanje	Da	Da
Brzina rada	9 600	9 600
Sopstvena serijska linija	–	Da
Podešavanje COM linije	–	–
Standardi		
T 4	Da	Da
T 30	Da	Da
CCIT V 21	Da	Da
CCIT V 27	Da	Da
CCIT V 29	Da	Da
Priloženi program(i)	BitFax	RFAX
Podržani graf. formati		
PCX	Da	Da
TIF	Da	Da
IMG	Da	Da
FAX	Da	Da

Korisne adrese

OSA
Miće Stojkovića 8, 11130 Beograd
Tel. 011/557-592, Fax 011/559-845

MIKRONOVA
Lenjinov trg 5/II, 21000 Novi Sad
Tel. 021/52-999, Fax 021/28-042)

WINDOWS 3.1 ANTE PORTAS

Nakon fantastičnog uapeha verzija 3.0 (uprkos nekim nedostacima ili nedoradanim opcijama, prodato ja preko 9 miliona primeraka), dugo očakivani *Windows 3.1* se, konačno, pojavio na tržištu, pa ćemo uskoro biti u prilici da procanimo koliko su ispunjana obećanja iz do sada nezabeležene pretpremijarna rakiamna kampanja. Za sada, da vidimo kako stvari atoja na prvi pogled.

Marko Kirić

Među dobrim poznavaocima sa smatra da su „okrugle" verzije velikih pakete zapravo finalne last verzija a da je prva prave verzija ona koja nosi oznaku x.1. Pakel *Windows 3.0* je jedan od sjajnih dokaza za ovo nezvanično pravilo koje se, uvidom u objavljeni spisak poboljšanja i novitele u gromoglasno najavljenoj i ambiciozno distribuiranoj „finalnoj" verziji 3.1 čini potpuno tačnim. Rad na novoj verziji atmosfera koja je oko nje stvorene i tržišna pripreme retko viđenog obima najbolje potvrđuju ovakvo pravilo. Tokom rade na novoj verziji je sprovedena ankata koja je obuhvatili nekoliko hiljade tipičnih korisnika, njihovi odgovori su detaljno razmotreni, a pri kraju posla ja angažovano čak 15 000 beta-testera.

Novi paket se istovremeno pojavio na šest jezika: engleskom, francuskom, nemačkom, španskom, italijanskom, holandskom i švedskom, i to istovremeno u SAD i Evropi (zapadnoj). Da bi se ovako ambiciozan cilj ostvario bez zastoja i problema, „Microsoft" je obučio i snabdeo dovoljnom količinom informacija preko 8 000 prodavaca i službenika koji rade na korisničkoj podršci, a zatim ja preduzeta najveća, vojnički rečeno, logistička operacija do sada viđena na tržištu softvera.

U okviru kampanje je aktivirana prodajna mreža i devet proizvodnih tačaka sa 24-časovnim radnim vremenom širom sveta, upotrebljeno je osam miliona disketa, količina paketa namenjenih samo američkom tržištu ja bila dovoljna da popuni 107 ogromnih kamiona-šlepera, a tokom kampanje se išlo čak i u takve detalje kakav je obezbeđivanje parkinga za zaposlene kako bi što manje gubili vreme! Rezultat se najbolje može iskazati kroz podatak da je 6. aprila ujutru više od 20 000 prodavaca širom sveta bilo snabdeveno sa više od 750 000 instalacionih kopija, dok je sam „Microsoft" pre toga već isporučio dodatnih 175 000 kopija *Upgrade* verzije samo u SAD.

Osnovni cilj je, naravno, bio maksimalno poboljšanje paketa i maksimalna prodaja. Da bi se to postiglo, prihvaćena je strategija rada u četiri glavna smera, to su lakoća korišćenja, brzina, pouzdanost i inovacije. Uneto je ukupno preko 1 000 izmena, većina kao rezultat najekstenzivnije saradnje sa korisnicima ikada viđene u kompjuterskoj industriji.

JEDNOSTAVAN RAD

Prvi kontakt korisnika sa paketom je instalacija: novi *Setup* prepoznaje još veći broj hardverskih i softverskih konfiguracija u odnosu na verziju 3.0, pri čemu je naročita pažnja obraćena na hardverske komponente i TSR programe za koje se znalo da često prave probleme. *Express Install* predstavlja metod za postavljanje osnovne konfiguracije i zahteva minimalno učešće korisnika, čime se početnicima olakšava instalacija. Naprednim korisnicima stoje na raspolaganju *Custom installation* opcije koje pružaju mnoštvo mogućnosti za fino podešavanje i maksimalno prilagođavanje potrebama, a za mreže stoji posebna *Network setup* procedura. Poređenja radi, inicijalna instalacija verzije 3.0 je obuhvatala 14 koraka, a u novoj verziji je svedena na samo 4, dok su podešavanja štampača i video-drajvera sa 5 svedeni na po tri koraka.

Windows 3.1 pokazuje svoju naklonost početnicima po on-line tutorijalu koji je stalno dostupan, a *Help* program je dramatično poboljšan kako u broju tema, tako i po dubini informacija. Korisniku sada stoji na raspolaganju i kontekstno osetljivi *Help*, koji se dobija prostim pritiskom na F1.

MENADŽERSKI POSLOVI

Novi *File Manager* je kompletno redizajniran i sada omogućava istovremeno prikazivanje stabla direktorijuma i njegovog sadržaja, podržava višestruke panele za pretraživanja po različitim drajvovima, omogućava prikazivanje većeg broja fajl-atributa nego ranije, kao i slobodan izbor fonta za prikazivanje naziva fajlova i foldera. Jedno od značajnih poboljšanja *File Manager*-a je svakako i nova **Quick Format** opcija, kojom se formatiranje drastično skraćuje. *File Manager* je snabdeven novim intuitivnijim „drag and-drop" modelom za manipulaciju fajlovima koji podrazumeva aktiviranja aplikacije i otvaranje fajla jednostavnim odvlačenjem ikone fajla na ikonu odgovarajućeg programa. Funkcionalnost ove opcije kontroliše takođe nova *Registration Database*, koja čuva informacije o načinu na koji aplikacije otvaraju i štampaju fajlove i grupiše tipove fajlova po pripadajućim aplikacijama.

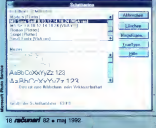
Program Manager takođe sadrži izvestan broj poboljšanja, od kojih su neka vidljiva odmah. Tako sada dugi nazivi ikona mogu da se protežu u više redova umesto jedne jedine neizgrapne linije, koja se često proteže preko susednih ikona. Tu je i nova *StartUp* grupa koja omogućava automatsko startovanje bilo koje grupe aplikacija automatski po inicijalizaciji *Windows*-a, a dodavanje u ovu grupu se obavlja jednostavnim odvlačenjem odgovarajuće ikona mišem.

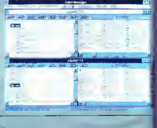
Printer Manager sada ima mogućnost nastavljanja posla koji je prekinut iz bilo kog razloga, bez intervencije korisnika. Ako, na primer, nestane papira, štampanje se automatski nastavlja po dodavanju nove količine. Drugo značajno poboljšanja predstavlja univerzalni printer-drajver UNIDRV koji predstavlja generički drajver za bilo koji štampač i koji omogućava jednostavnu nadgradnju za bilo koji model. Univerzalni drajver sadrži sva glavne parametre u jednom jedinom fajlu, a na proizvođaču je da obezbedi samo tabelu sa specifičnim parametrima. Umesto korišćenja nekoliko tuceta posebnih drajvera, sada je dovoljan samo jedan opšti drajver i *support*-tabela za svaki pojedinačni printer. UNIDRV podržava preko 260 modela štampača.

BRZINA

Prema „Microsoft"-ovim specifikacijama, ubrzanje u odnosu na verziju 3.0 se ogleda u pet aspakata:

• Komponente grafickog okruženja tj. kon-





Microsoft Photo Service



Microsoft Photo Service

snićkog interfejsa – *File Manager* i *Program Manager*

- Keširanje diska – *SmartDrive* je kompletno redizajniran i automatski se instalira prilikom postavljanja sistema omogućava keširanje i pisanja i čitanje Prema specifikacijama ubrzanje u odnosu na staru verziju istog programa zinosi 10-100% *Windows 3.1*, inače, podržava 32-bitni pristup disku, što omogućava zaobilaženja DOS-a na nivou BIOS-a i dodatno ubrzanje diska
- Brže preplitanje (*paging* – straničenje) u 386 (*enhanced*) modu
- Poboljšanim video-drajverima kao što su VGA i 8514
- Povećana brzina štampanja uz brži povratak kontrole izvornoj aplikaciji po završenom štampanju Dodatno ubrzanje se postiže dinamičkim *download*-om fontova

POUZDANOST

Korisnici *Windows 3.0* vrlo dobro znaju na šta se ova kategorija odnosi – retki su oni koji nikada nisu dobili poruku **UAE – Unrecoverable Application Error** Ovaj problem je, izgleda, shvaćen ozbiljno pa su preduzete četiri vrste preventive

- Zaštita samog okruženja preko procene parametara (*Parameter validation*) Reč je o novoj internoj opciji koja provodi za korisnika potpuno transparentnu proveru i procenu parametara za alokaciju resursa, hendlere i pointere Naime, bez takve proveru, *Windows* aplikacija bi mogla da zahteva od operativnog sistema da izvrši nekorektnu alokaciju kao što je pisanje preko važnih sistemskih podataka u memoriji ili slanje podataka na pogrešnu adresu U verziji 3.0, ovakve greška su dovodile do nepredvidivog ponašanja same aplikacije gubitka podataka ili kraha sistema Mehanizam procene parametara u verziji 3.1 sve takve zahteve treba da vraća aplikaciji koja ih je poslala kao neizvršene
- Oporavak od aplikacionih grešaka – ozbiljna aplikaciona greška je u verziji 3.0 proglašavana „nepopravljivivom aplikacionom greškom" (UAE), i rezultirala izlaskom iz aplikacije, ostavljajući korisnika bez informacija o tome šta se zapravo desilo uz preporuku da resetuje sistem kako bi predupredio moguće posledice Verzija 3.1 u takvom slučaju identifikuje izvor greške, omogućava korisniku da ili odmah izađe iz aplikacije ili se vrati radi snimanja podataka i na kraju pokušava da održi sistem u radnom stanju i posle greške
- Resetovanje aplikacija (*Application Reboot*) označave mogućnost izlaska iz blokirane *Windows* ili DOS aplikacije pritiskom na Ctrl-Alt-Del pri čemu se korisniku nudi mogućnost zatvaranja aplikacije, povratka u sistem ili reseta celog sistema.
- Obezbeđivanje alata i podrške za testiranje nezavisnim proizvođačima softvera kako bi bili u stanju da razviju stabilnije *Windows* aplikacije

OBJEKTI, FONTOVI I DRUGE PRIČE

Jedna od najvažnijih tehnologija devedesetih je OLE – *Object Linking and Embeding* – koja omogućava kreiranje okruženja u kojima aplikacije mogu da razmenjuju informacije bez prepreka Reč je o pristupu svim podacima kao o objektima, bez obzira da li je reč o grafikonu, ilustraciji tabeli ili tekstu – sve su to objekti *Windows 3.1* podržava OLE tako što obezbeđuje standardne biblioteke, interfejse i protokole koje koriste aplikacije pri razmeni objekata. „Microsoft" je snabdeo OLE mogućnostima nove verzije programa koji se nalaze u sastavu paketa – *Write Paint* i *Cardfile*

Windows 3.1 konačno koriste *TrueType* tehnologiju fontova promenljive veličine (*scalable fonts*) ova tehnologija se zasniva na konturama fontova iz kojih se generišu sva potrebne veličine i načini prikaza. *TrueType* tehnologija je projektovana i razvijena za profesionalce i grafičke dizajnere ali tako da istovremeno bude dostupna svima i laka za upotrebu Pored svih lepih osobina, najvažnija je svakako potpuni WYSIWYG (*What You See Is What You Get*) – otisak na bilo kom printeru od matričnog do *PostScript* lasera, se savršeno poklapa sa ekranskim prikazom, i uz to eliminiše potrebu za posebnim skupim kartridžima.

Oni koji ne mogu bez DOS-a su takođe došli na svoje – *Windows 3.1* nudi jaču podršku DOS-aplikacijama – podržane su aplikacije koje rade u VGA grafičkom modu tako što im je omogućen rad u prozoru ili u pozadini, kao i podrška miša. Osim toga, paket sadrži i veliki broj PIF fajlova koji sadrže informacije potrebne za rad DOS aplikacija pod *Windows* okruženjem Konačno 32-bitni pristup disku omogućava istovremeni rad sa više DOS aplikacija nego što je to bilo moguće u verziji 3.0

Pored navedenih novosti tu je još i ceo niz krupnijih i sitnijih poboljšanja, izmena i potpuno novih mogućnosti kao što su podrška za Multimediju, laptop računare i sve popularnije pen-računare

PROZORI I OLOVKE

Olovka nude jedinstvene mogućnosti – ona je svima dobro poznata, prirodna, i može se koristiti sedeći, stojeći ili ležeći. Pored toga. olovka je i idealan grafički instrument – možete pisati ili crtati slobodnom rukom, konstruisati dijagrame, mape, slike i još mnogo toga. Personalni računari, sa druge strane, evoluiraju ka manjim, prilagodljivijim formama, tako da sve više počinju da liče na današnje notese ili džepne rokovnike nego na klasične stone skalamerije Elektronske tablice će omogućiti sasvim novim kategorijama korisnika (uključujući i raznosače pošte pronalazače prodavce i mnoge druge) da koriste olovku koriste i računar Kombinacijom toga dvoga, dolazimo do lakšeg hvatanja beleški bolje organizacije vremena i jednostavnije primene računare na javnim mestima.

Bazirani na *Windows 3.1*, *Windows for Pens* predstavlja originalni paket koji se isporučuje sa mnogim novim računarima ovog tipa. Sa ovim paketom, rukopis se može sačuvati takvim kakav jeste, u izvornom obliku ili u obliku otkucanih karaktera. „Olovkom" se istovremeno mogu i zadavati komande kao što se to radi mišem na stonom računaru Da bi to sve bio moguće potrebno je da računar odnosno operativni sistem prepoznaje i podržava specifične operacije – *Windows for Pens* raspolaže drajverima za operacije kao što su detekcija pisanja i „mastila", standardnim setom od 13 sistemskih gestova, podrškom za korisnicki definisane gestove i gestove definisane u aplikacijama, podršku za prepoznavanje više različitih rukopisa, kao i podrškom za čisto pisane „normalne" *Windows* aplikacije.

Na osnovu svega što smo do sada čuli i videli, *Windows 3.1* bi trebalo da opravdaju buku i iščekivanje kojima su bili okruženi proteklih meseci i postanu sledeći važan korak u „Microsoft"-ovoj razvojnoj strategiji Ako ništa drugo, sigurno će nam pomoći da strpljivije sačekamo pojavu i punu operacionalizaciju *Windows NT*

MARŠ SVI NA UNIX!

Kad se povede razgovor o operativnom sistemu UNIX i njegovim mogucnostima, čak se i danas, ne samo među laicima nego i programerima, može se čuti pitanje: „Čemu sve to, kada postoji DOS i Windows?". Na ovo pitanje nije lako odgovoriti – prosto zato što nabrajanje razloga suviše dugo traje.

Aleksandar Dumnjanović

Današnje „savremeno" rešenje na nekoj 386 ili 486 mašini sa DOS ekstenderima, posebnim programima za keširanje rada sa diskom, uz gomilu raznoraznih programa koji se međusobno slabo podnose i svaki za sebe smatra da mora da vodi računa o svemu što se na mašini dešava, nema nikakvih šansi da omogući puno iskorišćavanje hardverskih resursa (o pouzdanosti da i ne govorimo).

Usklađivanje svih, pre svega softverskih, komponenti sistema postalo je danas prava noćna mora i otvara vrata novoj profesiji – ljudima koji na prvi pogled mogu da u konfiguraciji sistema otkriju „neprihvatljive" kombinacije programa koja ugrožavaju pouzdanost sistema.

UNIX ne boluje od ovakvih problema, jer, prosto, u sebi već sadrži one funkcije koje se teško „kaleme" na operativni sistem: virtuelnu memoriju sa straničenjem po zahtevu (engl. „demand paging"), keširanja rada sa diskom, multitasking itd.

ČEMU JE NAMENJEN?

UNIX ima više lica: koristi se na grafičkim radnim stanicama na kojima možda nikada nećete videti komandnu liniju, za knjigovodstvene poslove, gde su alfanumerički terminali priključeni serijskim linijama na centralni računar, za „number-crunching" na superračunarima koji uopšte nemaju direktnu vezu sa terminalima itd.

Jedna tipična konfiguracija predstavlja klasičan višekorisnički sistem sa neinteligentnim alfanumeričkim terminalima povezanim serijskim vezama (obično RS232) sa centralnim sistemom. Ova varijanta daje dobar odnos cena/performanse i demonstrira jednu posebnu pogodnost UNIX-a: on može da radi sa praktično svakim terminalom, jer se svi programi koji kontrolišu ekran oslanjaju na opis terminala koji se zadaje u posebnoj datoteci.

Za ove primene uobičajena je upotreba inteligentnih komunikacionih kontrolera koji značajno rasterećuju centralni procesor. Pri tome se dobitak ostvaruje samo pri slanju – prikazu na terminalima i slanju podataka na štampače, u obrnutom smeru se često uopšte ne postiže ušteda u opterećenju procesora, jer je uobičajeno da aplikativni program reaguje na svaki pritisak tastera na terminalu. Opterećenje procesora vezano za komunikaciju sa terminalima često može da predstavlja realno ograničenje broja radnih mesta.

Korisnicima naviklim na DOS okruženje može da smeta relativno niska brzina prenosa podataka serijskom linijom (tipično do 38400 bps efektivno i znatno manje) što onemogućava trenutne promene sadržaja ekrana koje se na personalnom računaru mogu postići direktnim upisom u memoriju video kartice. Pored toga, većina terminala ne može da prikaže sve karaktere koji se koriste u DOS aplikacijama, pa izgled ekrana na terminalu obično nije tako „doteran" kao kod DOS-a. Većina terminala, takođe, nema ni PC-kompatibilna tastature, pa se programeri ne mogu oslanjati na razne kombinacije tastera na koje su navikli na personalnim računarima.

Ovakvi sistemi se koriste za različite aplikacije, od knjigovodstvenih do inženjerskih, gde je komunikacija sa korisnicima u tekstualnom režimu.

Po potrebi mogu biti povezani u mrežu sa drugim računarima.

Druga tipična vrsta UNIX sistema je grafička radna stanica. Obično je priključena na lokalnu računarsku mrežu i, pored toga što ima sopstveni podsistam diskova, oslanja se na neki veći računar kao server za datoteke. Koristi X Window sistem – standardni sistem za upravljanje prozorima na UNIX-u, pri čemu se na radnoj stanici izvršava X server, a X klijenti (aplikacije) se mogu izvršavati na drugim računarima u mreži (na taj način se ovi drugi računari pretvaraju u „procesorske servere"). Ovakvi sistemi su namenjeni prvenstveno za CAD i druge primene koje zahtevaju izuzetne grafičke mogućnosti.

Ispod aplikativnog sloja softvera, koji mora biti različit za ovako različite platforme i namene, nalazi se, u suštini, ista podloga. Za svaku od ovih varijanti sigurno se može napraviti bolje rešenje koje je optimalno podešeno za konkretan računar i konkretnu primenu, međutim praktično je nemoguće nadoknaditi veliku prednost koju UNIX ima zato što je već dugo prisutan i brzo se može preneti na svaki novi računar.

HARDVERSKA NEZAVISNOST

Poseban je doživljaj kada sednete pred neku novu mašinu, poslednju reč računarske tehnologije, za koju (naravno) niste imali prilike da vidite ni jedan list dokumentacije, ali se već posle nekoliko minuta snalazite „kao kod kuće". Ovako nešto ne važi samo za sisteme sa „službenim nazivom" UNIX. Svaki poznavalac UNIX-a sasvim će se dobro snalaziti i na AT računaru pod operativnim sistemom XENIX/286 i na IBM RS6000 RISC mašini pod operativnim sistemom AIX. Treba napomenuti i da je, kada se pojavi novi procesor na tržištu, praktično jedina šansa da se računari sa ovim procesorom na vreme opreme softverom upravo operativni sistem UNIX – zato što je skoro čitav pisan u programskom jeziku C. I zato što je njegov izvorni kod moguće dobiti (izvorni kod AT&T UNIX-a se „debelo" plaća, a neke verzije Berkeley UNIX-a su i danas besplatne). Potrebno je samo razviti C kompajler i iznova napisati delove operativnog sistema koji su specifični za konkretnu arhitekturu. Razvoj C kompajlera nije tako komplikovan kao što zvuči – na UNIX-u danas postoji čak i izuzetno kvalitetan PD kompajler za C/C++ – GNU C – kojem se formalnim jezikom opisuje procesor za koji treba da generiše mašinski kod.

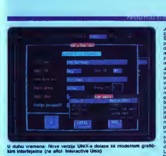
Postojanje velikog broja različitih modela računara na kojima se UNIX „vrti" i varijanti UNIX-a tera proizvođače softvera na neuporedivo veću pažnju pri programiranju nego što je to slučaj kada je softver isključivo namenjen jednom tipu računara. Nije neobično videti da se neki softver može kupiti za npr. 60-tak varijanti UNIX-a, za najrazličitija mašina – nije suviše teško održavati sve ove varijante uz odgovarajuću pažnju i disciplinu. Moraju se predvideti različiti načini smeštanja binarnih brojeva (zavisno od toga da li se prvo smešta bajt najveće važnosti, kao npr. kod Motorolinih procesora, ili najmanje važnosti, kao npr. kod Intelovih procesora), podrazumevana veličina binarnog broja (ovo je „prirodna" veličina tipa „int" za konkretnu mašinu – u današnje vreme je uobičajena veličina od 4 bajta – za „32-bitne" mašine), veličina ukazatelja (pointera), da li mašina „prirodno" koristi označene ili neoznačene bajtove itd. Pored toga, mogu postojati neke razlike i u bibliotečkim funkcijama. Uprkos svemu tome, prenos (engl. „port") izvornog koda postojećeg softvera sa jedne na drugu varijantu UNIX-a ne predstavlja veliki problem – neuporedivo manji nego prenos na drugi operativni sistem.

MULTITASKING

UNIX je, od samog svog davnog početka, bio pravi multitasking sistem. To znači da je za UNIX istovremeno izvršavanje više programa ne samo normalna stvar, već i suštinska osobina bez koje mnogi postojeći programi ne bi ni mogli da rade. Pri tome su svi programi „zaštićeni" jedan od drugoga, za UNIX ne moraju programirati samo najbolji programeri sveta. Fatalni kraj jednog od više desetina programa koji se istovremeno izvršavaju ostale neće ni najmanje uzbuditi (u najvećem broju slučajeva, ostali to neće ni primetiti). Većina „ozbiljnijih" aplikativnih paketa na UNIX-u koristi više procesa u jednom trenutku. Ovim se može postići podela na funkcionalne celine koje se odvojeno razvijaju i testiraju, a pojedini delovi se lakše zamenjuju ili kombinuju. Osim toga, mogu se praviti specijalizovani „serveri", tj. procesi koji za sve korisnike u sistemu obavljaju određene poslove: to se u mnogim operativnim sistemima rešava proširivanjem jezgra operativnog sistema (engl. „kernel"), što kod UNIX-a nije obavezno. Tekući pravac razvoja operativnih sistema ide upravo na tu stranu: izmešta se što je moguće veći deo funkcija koje su se tradicionalno rešavale u jezgru operativnog sistema u posebne serverske procese, dok u jezgru ostaje samo nužni minimum. Prema onome što se može pročitati u literaturi, takvi operativni sistemi su QNX, *Windows NT* i famozni *Plan 9*, potencijalni naslednik UNIX-a iz „Bell" laboratorija.

VIŠEKORISNIČKI SISTEM

Pored zaštite memorijskog prostora programa koji se izvršavaju od, namernog ili slučajnog, „upada" sa strane, UNIX pruža i zaštitu podataka korisnika sistema. U pitanju je uglavnom klasičan metod zadavanja prava pristupa datotekama i uređajima, uz korišćenje pseudonima i lozinki pri prijavljivanju korisnika. Sistem obično omogućava i registrovanje upotrebe re-



U duhu vremena: Nove verzije UNIX-a dolaze sa modernim grafičkim interfejsima (na slici: Interactive Unix)

sursa za svakog korisnika ponaosob radi kontrole ili naplate. Prava pristupa (i mnoge druge stvari) određuje tzv. „superkorisnik", odnosno administrator sistema. Ovaj sistem ima smisla i za radnu stanicu, tj. sistem sa samo jednim radnim mestom – svaki korisnik može imati svoj „deo diska" na kome je bezbedan od ostalih. Čak i za sistem sa samo jednim korisnikom dobro je imati osim administratorskog i jedan „običan" pseudonim – jer greška u kucanju komandi koje učinite kao administrator mogu mnogo da koštaju.

MREŽE

UNIX sistemi se mogu povezivati u mreže na različite načine, počevši od veza direktnim serijskim linijama, preko modemskih veza do brzih, lokalnih ili čak udaljenih veza (ovo poslednje je za nas samo fantastika). Ove veze se mogu koristiti za prenos elektronske pošte, prenos datoteka, deljenje datoteka u mreži (engl. „file sharing"), priključivanje korisnika na udaljene računare, sve do X Window sistema u kojem se može otvoriti više prozora na ekranu radne stanice, i u svakom prozoru pokrenuti program na nekoj drugoj mašini u mreži. Ne treba zaboraviti da je TCP/IP (komplet protokola za komunikaciju u mrežama) nastao upravo na UNIX-u.

Tek sa talasom „demokratizacije virtuelne memorije" koju donose jeftini 386 procesori (i njihovi klonovi) UNIX počinje da konkuriše operativnom sistemu Novell i po ceni kompletnog rešenja. Za razliku od popularnog Novell-a, UNIX sistemi u mreži mogu biti istovremeno i klijenti i serveri. Pri tome na svaki od sistema se preko serijskih linija može priključiti više terminala, pa je rešenje jeftinije za isti broj radnih mesta. Zbog „specijalizacije" servera pod Novell-om, on daje bolje rezultate, pa su se u skorije vreme pojavile varijante sprege UNIX sistema i Novell servera u lokalnoj mreži.

STABILNOST SISTEMA

Daleko bi od istine bilo reći da „UNIX ne pada", tj. da na računarima sa operativnim sistemom UNIX ne dolazi do zaglavljivanja sistema ili drugih incidenata. Doživeo sam to više puta na vrlo različitim verzijama UNIX-a, i na vrlo različitim računarima. Prema mom iskustvu, u praktično svim slučajevima razlog je bio u loše napisanim veznim programima (engl. „driver") ili u nepouzdanom hardveru. Što se tiče PC-kompatibilnih računara, došao sam do (za mene neverovatnog) zaključka da se u mnogo slučajeva za „padove" pod DOS-om pogrešno krivi, istina nepouzdan i nestabilan softver, umesto lošeg hardvera. Posebno je teško otkriti pravi uzrok problema kada se ovi javljaju retko, npr. jednom dnevno ili ređe. U UNIX okruženju nešto je lakše pronaći krivca: uglavnom je u pitanju podsistem diskova, zato što na probleme sa drugim periferijama UNIX nije ni blizu toliko osetljiv.

Da se ovaj odeljak ne bi zvao „nestabilnost", moram da dodam: ako se izuzmu problemi za koje su krivi proizvođači hardvera (oni isporučuju i vezne programe), UNIX je vrlo pouzdan i stabilan sistem. Za ovo treba zahvaliti pre svega hardverskoj zaštiti memorije, koja je postala standard, kao i upravo ogromnoj količini truda koja je uložena u UNIX – jednostavno, greške u jezgru su odavno „istrebljene" – a ostale, i ako ih ima, ne mogu biti tako opasne. Ako ga previše opteretite, on se neće zaglaviti: samo će raditi sporo, i (eventualno) prikazati na ekranu glavnog terminala poruku o prekoračenju veličine neke sistemske tabele.

OTVORENOST

UNIX je, u svojoj osnovi, veoma jednostavan. Broj sistemskih poziva je neuporedivo manji nego na primer kod DOS-a, pri tome, svi su oni dokumentovani. Nije nikakav problem napisati komandni procesor za UNIX, potrebno je samo vremena i strpljenja za čitanje dokumentacije. Maternji jezik UNIX-a je C: kada napišete svoj komandni procesor, vaš program će razumeti svaki C-prevodilac na bilo kojoj UNIX mašini, i posle prevođenja će se normalno izvršavati. Nisu bitne razlike u tipu procesora, arhitekturi računara, proizvođaču operativnog sistema ili prevodioca.

Multitasking i mogućnost komunikacije među procesima vrlo se dobro mogu iskoristiti već i na nivou komandnog procesora, tj. sa komandne linije. Zahvaljujući tome, mnoge komande operativnog sistema, kao i dobar deo inicijalizacije („podizanja") sistema često su realizovani kao komandna procedura. Novajlije na UNIX-u, kad se malo „prošunjaju" po sistemu, često se čude kako je „sve tako jednostavno". Naravno, jednostavno je i na drugim operativnim sistemima, samo što se ovde mnogo više toga vidi!

RAZVOJNO OKRUŽENJE

Tradicionalni jezik za razvoj softvera na UNIX-u je C: ogromna većina prevodilaca za druge programske jezike generiše kod koji se može (ili mora) linkovati sa bibliotekama C-a. Pri tome nema drastičnih skokova, kao na DOS-u, gde je uobičajeno da nova verzija prevodioca nije kompatibilna sa starom verzijom od istog proizvođača. U novije vreme se pojavljuje migracija na ANSI standard jezika C, koja je na UNIX-u znatno zakasnila u odnosu na DOS. Pošto je ogromna većina softvera na UNIX-u pisana u UNIX C-u, neki proizvođači danas nude dva odvojena prevodioca – jedan tradicionalni, i jedan koji podržava ANSI standard. Može se očekivati da će ovde biti određenih problema: ja ni danas ne koristim ANSI C, jer je programiranje u „originalnom" C-u još uvek portabilnije. U poređenju sa prevodiocima na DOS-u, dodatni alati izgledaju znatno skromnije: programeri uglavnom koriste komandnu liniju, a ako su im potrebni prozori, otvaraju ih sami, u X Window sistemu. Na raspolaganju je uvek neki simbolički dibager (program za otklanjanje grešaka – engl. „debugger"), ali se uglavnom po udobnosti upotrebe ne može takmičiti sa sličnim alatima na DOS-u. Značajna prednost razvoja na UNIX-u je u zaštiti memorije – kada program „promaši" svoj adresni prostor, i još u nekim situacijama, automatski se formira kopija njegovog adresnog prostora na disku, tako da se pomoću simboličkog dibagera može izvršiti „post-mortem" analiza. Za razvoj softvera koji analizira bilo kakav ulazni tekst (prevodioci, interpreteri, svi programi koji imaju „komandnu liniju") nezaobilazan alat je poznati YACC, generator parsera. Za razvoj softvera koji radi u *X Window* sistemu može se kupiti razvojni paket za X: on sadrži, pre svega, biblioteku potprograma za pisanje tzv. X klijenata. O tome ću reći samo onoliko koliko sam „iz aviona" uspeo da vidim: po glomaznosti se može porediti sa programiranjem za *Windows* okruženje.

Pored standardnih razvojnih alata, naravno, mogu se kupiti i drugi prevodioci za sve moguće jezike, razni sistemi za upravljanje bazama podataka, CASE alati itd. Svega ima u izobilju, samo treba platiti.

APLIKATIVNI SOFTVER

UNIX je „domaći teren" za ogroman broj izuzetno kvalitetnih aplikativnih programa za najrazličitije namene („izuzetno" u odnosu na uobičajen nivo za DOS): dva glavna razloga što su oni nedovoljno poznati kod nas su u njihovoj obično visokoj ceni i u velikim zahtevima koje postavljaju u pogledu resursa sistema. Ovo nije apsolutno pravilo, ali za luksuzni softver treba platiti luksuznu cenu i „voziti" ga na moćnoj mašini.

BUDUĆNOST

UNIX i njemu srodni operativni sistemi već su dugo prisutni i razvijaju se i dalje, pre svega zahvaljujući konkurenciji među proizvođačima softvera. Pored toga, UNIX je najrasprostranjenija platforma na američkim univerzitetima, gde se stalno razvija i poboljšava; neka od tih poboljšanja na kraju nađu mesto i u komercijalnim verzijama. Nijedan drugi operativni sistem koji je vlasništvo bilo koje firme ne može da računa na toliki broj visokokvalifikovanih programera koji uz to rade besplatno (delimično uz finansijsku pomoć države za istraživanja). POSIX standard za korisnički i programerski interfejs operativnog sistema najbliži je UNIX-u, „privođenje" UNIX-a ovom standardu je već u toku.

Na području manjih i srednjih višekorisničkih sistema UNIX je stekao vrlo snažne pozicije i praktično i nema konkurencije (konkurenti su tek najavljeni): u oblasti grafičkih radnih stanica bi situacija mogla da se komplikuje sa pojavom novih operativnih sistema „Microsoft"-a i IBM-a, naročito za jeftinije sisteme. Posebno će biti interesantno ako se daljim padom cena hardvera dođe do napuštanja neinteligentnih alfanumeričkih terminala u korist slabih grafičkih radnih stanica.

SVE SVOJE SOBOM NOSI

Da nama ništa zdravije od zdrava konkurancija pokazuja i najavažiji primar iz trka dva najveća proizvođača C kompajlera. Možda se Borland C++ 3.0 nikada na bi pojavio u ovako impresivnom izdanju da mu Microsoft nija stalno bio za petama.

Zoran Životić

U paketu ćete pronaći sve što da vam ikada od alata zatrebati za ozbiljan razvoj programe u DOS i *Windows* okruženju. Spisak je zaista impozantan – od kompletnog C++ razvojnog okruženja za DOS, preko asemblera, profajlera i dibagera, do potpuno ponovljenog C++ okruženja, ovoga puta za *Windows*, sa odgovarajućim dodatnim alatima. Uz sve to dobijate i oko 3 000 strana uputstva, uz oko 30 MB podataka i programa na disku.

PAKET I INSTALACIJA

U paketu BC++ 3.00 nalazi se pet programskih celina:

- Borland C++ 3.0, DOS i *Windows* verzija C++ kompajlera sa odgovarajućim razvojnim integrisanim okuženjima i dodatnim alatima za DOS. Uputstvo čine četiri knjige – vodič za korisnika (230 str), vodič za programera (470 str), vodič za alate (250 str) i referentni priručnik za biblioteke (650 str)
- *Resource Workshop* – program namenjen kreiranju svih elemenata koje čine *Windows resource* datoteku (uputstvo str. 290)
- *Turbo Profiler 2.0* kojim se obavlja detaljna analiza efikasnosti pojedinih delova vašeg programa (uputstvo, str. 220)
- *Turbo Assembler 3.0* (uputstvo, str. 380, kratki podsetnik str. 150)
- *Turbo Debugger 3.0* (uputstvo, str. 470)

U paketu se nalaze i diskete u oba formata – 3.5" i 5.25". Diskete nisu odvojene po pojedinim programskim celinama, već samo numerisane brojevima – sve je stalo na po 9 disketa u svakom od formata.

Instalacija se obavlja na manje-više klasičan način – pokretanjem *Install* programa sa prve diskete nudi vam se meni za izbor pojedinih opcija, gde možete postaviti odredište za instalaciju i izabrati instaliranje samo nekih od elemenata paketa. Ukoliko se opredelite za potpunu instalaciju, čitav proces kopiranja i raspakivanja (datoteke na disketama su komprimovane poznatim ZIP arhiverom) će potrajati oko 20 minuta (386/33, 4MB, „Seagate" IDE disk). Instalacija je, kako već počinjemo da se navikavamo u poslednje vreme, prilično bolan proces – potrebno je na disku obezbediti oko 30 MB slobodnog prostora. Ako ste, kao i ja, „tanki" sa slobodnim prostorom, instalacioni program vam neće ni malo pomoći – umesto sumarnog prikaza potrebnog prostora s obzirom na izabrane opcije, *Install* samo informativno daje dužine pojedinih delova, tako da vam ostaje da uz papir i olovku kalkulišete da li se uklapate u ono što vam je na raspolaganju. Procenjujući onako „od oka" da bi 25 Mb trebalo da bude dovoljno, pokrenuo sam instalaciju i dobio upozorenje da slobodan prostor na disku *možda* neće biti dovoljan. Kada računar ispiše „možda..." onda je zaista teško propustiti priliku da se proveri koliko je „ugrađena inteligencija" u pravu – na moju nesreću procena je bila dobra i instalacija je zaista prekinuta negde pred kraj, kada su već sve datoteke uveliko bile kopirane a delom i raspakovane.



Oslobodivši uz krajnje napore još dodatnih 5 MB, ponovo sam dobio isto upozorenje, ali sam ipak odlučio da ponovo testiram procenu. Ovaj put je, na sreću, bila pogrešna. Kompletna instalacija je završena uspešno, ostavljajući čak oko 2MB slobodnog prostora. Iako je ovakav paket namenjen profesionalnom razvoju i pretpostavlja jaču platformu, ipak verujem da su čak i među njima retke one sa diskovima većeg kapaciteta od tipičnih 80-120 MB – u situaciji kada treba trećinu ili četvrtinu diska odvojiti samo za jedan paket, *Install* program bi ipak mogao malo više da se potrudi da nam pomogne od lakonske izjave „to možda neće moći...".

WINDOWS OKRUŽENJE

S obzirom da je pojava svakog novog razvojnog orkuženja sa *Windows* zaista novost, prvo me je zanimalo kako izgleda taj deo BC paketa. Kao poređenje ću koristiti *Microsoft QC/Win* pošto vam je verovatno još u sećanju prikaz iz prošlog broja „Računara". Nakon instalacije, koja je inače u potpunosti protekla u DOS okruženju, pri prvom sledećem startovanju *Windows*-a, dobijate pitanje da li želite da instalirate BC++ grupu sa programima. Ukoliko koristite neki od alternativnih *shell* programa, treba ga pre startovanja skinuti i vratiti originalni *progman.exe*. Iako koristimo *Norton Deskop* i bez ove operacije sve je proteklo u najboljem redu i nakon potvrdnog odgovora na pitanje dobili smo grupu BORLAND C++ sa deset ikona. Startovanjem prve sa oznakom TURBO C++ ulazimo u integrisano okruženje za razvoj programa.

Okruženje je, u osnovi, ono što se pod *Windows*-ima očekuje. Dobijate osnovni prozor koji čini „vaš svet" – unutar njega se može otvoriti više prozora sa odvojenim modulima (klasičan MDI interfejs), a ispod osnovnog menija se nalazi i niz ikona za brzo pokretanje pojedinih češće korišćenih opcija („Borland" ovaj deo naziva **Speed Bar**, za razliku od uobičajenog termina **Tool Bar**). Nakon učitavanja nekog programa, prvi utisak koji odaje editor je brzina. Dok nismo probali ovaj editor, smatrali smo da se pod *Windows*-ima i ne može postići neka posebna brzina pomeranja teksta – editori koje sam imao prilike da probam (od NOTEPAD-a, preko QC/WIN do WINWORD-a) nisu baš bili impresivni u pogledu brzine. Iako se, naravno, ovaj editor ne može ni približno meriti sa klasičnim editorima u tekst režimu, niti je razlika u odnosu na „Microsoft" spektakularna, ipak je interesantno da se i pod *Windows*-ima može dobiti na brzini odgovarajućim kodiranjem.

Editor nema opciju za prikaz pojedinih elemenata programa različitim bojama, što je zaista šteta – takva opcija neverovatno doprinosi preglednosti programa – ali mu zato ne nedostaje mogućnosti za različita podešavanja. Od izbora iz tri različita seta komandnih tastera, preko podešavanja tabulator opcija, izbora pisma za prikaz teksta, do takvih detalja kao što je položaj pomenute konzole sa alatima, koja može biti ne samo vertikalna ili horizontalna, već i ubačena u sopstveni prozor, koji se onda slobodno može pomeriti na proizvoljno mesto. Kao i kod QC/Win, ni ovaj editor nema mogućnost definisanja sopstvenih komandnih tastera. Ono što mi je ipak posebno nedostajalo je skraćeni način za prelazak iz jednog prozora u drugi – pored Ctrl Tab kombinacije koja „vrti u krug", nema načina da jednim potezom pređemo na proizvoljan prozor.

Proces rada na nekom programu počinje otvaranjem odgovarajućeg projekta i specifikacijom datoteka koje u njega ulaze. BC podržava generisanje svih poznatih formata – od *Windows exe*, preko DLL biblioteka do „Borland"-ove verzije „DOS/Windows" kombinacije nazvane *EasyWin*, kojom se „obični" DOS programi prevode bez ikakvih izmena u nešto što liči na *Windows* aplikacije (zapravo se **stdin/out** usmerava na klasičan *Windows* prozor). Pre generisanja programa, potrebno je podesiti niz opcija kojima se utiče na proces prevođenja. Možete se ograničiti samo na K&R C mogućnosti, ili tražiti potpunu ANSI ili Unix V kompatibilnost i, naravno, sam BC++ režim. Kod može biti generisan u sva četiri poznata memorijska modela, a na njegovu kompaktnost i brzinu možete uticati takođe prilično klasičnim setom opcija za optimizaciju. Dva „Borland" specijaliteta se ipak kriju među opcijama kompajlera – jedan poznat od ranije je ukidanje duplih statičkih stringova (sve poruke koje ispisujete kodirajući ih u programu da bivši uporeođene i duplikati izbačeni), i drugi koji je apsolutna novost i verovatno da imati uticaja i na ostale



proizvođače kompajlera – pretkompajlirana zaglavlja

Ukoliko ste nekada analizirali čime sve kompajler treba da se pozabavi u prevođenju programa, sigurno ste primetili da zapravo najveći deo rada sa diskom otpada na učitavanja i obradu često velikog broja #include datoteka, čiji ukupan broj linija (posebno kada se tu nađe **windows.h**) uveliko prelazi dužinu vašeg programa. Ideja da se programi razbijaju na što sitnije module time je znatno degradirana, jer teramo kompajler da troši vreme na obradu nekoliko hiljada, i to uvek istih, linija koda samo da bi potom obradio naših stotinjak u svakom od modula. „Borland" je pronašao odlično rešenje za ovaj problem – sve .H datoteke koje uključujemo u projekat jednom se kompajliraju u interni .SYM format i potom koriste direktno u toj formi. Rezultat je *drastično* ubrzanje procesa prevođenja – odnos je teško i izmeriti, jer obrada pretkompajliranih zaglavlja traje izuzetno kratko.

Metod ima i svoja ograničenja – zaglavlja ne smeju sadržati informaciju koja bi se prevela u kod, moraju se nalaziti u svim modulima u istom rasporedu i moduli moraju biti kompajlirani istim opcijama (samo ograničen skup, ne sve opcije). Ova ograničenja su sasvim razumna i u praksi ne predstavljaju nikakav problem, tako da je ova novost jedna od onih prijatnih „sitnica" koje nas je stvarno obradovala.

BORLAND LOOK & FEEL

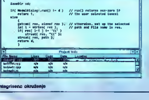
Podešavajući opcije kompajlera, primetićete svakako i „Borland look" dijalog – umesto standardnog *Windows* grafičkog dizajna, „Borland" je napravio potpun redizajn svih elemenata – površina dijalog prozora su u raster sivoj varijanti sa kontinualnim sivim površinama, na kojima su raspoređena *edit* i *radio* kontrole. Ni dugmad nisu *Windows* dugmad – znatno su krupnija od uobičajenih, a simbol ostaje u prednjoj ravni i onda kada se dugme uvuče usled pritiska. Dizajn je, naravno, pitanje ukusa, ali verujem da će se mnogi složiti da se ipak radi o nepotrebnom opterećivanju ukupnog izgleda koji sada postaje znatno zasićeniji (svi dijalozi zapravo deluju pomalo „mračno"). S obzirom da u osnovnom dizajnu prozora ništa nije menjano i da pred očima najčešće imamo svetlu i čisto belu sliku, iskakanje tamnih, zasićenih maski deluje malo zamorno za oči. Kada se ovome dodaju među-ekrani sa crtežima koji se pojavljuju dok traje učitavanje programa (nakon startovanja, recimo, *Workshop*-a dobijate sliku devojke koja u idiličnoj atmosferi neke radionice koristi bušilicu – posle toga ipak pre očekujete da se pojave Lemingzi nego ozbiljan alat) ipak nećete moći da se otmete utisku da je ceo dizajn pod snažnim uticajem američkog shvatanja dobrog ukusa. O tome, kao što rekosmo...

Kada ste podesili sve opcije, možete konačno pokrenuti prevođenje programa – na ekranu će se pojaviti prozor sa osnovnim podacima o toku procesa. Kao i kod QC/Win, i ovde se može primetiti nešto duža pauza pre nego što, naizgled, išta počne da se dešava. Verovatno se radi o vremenu potrebnom za analizu zavisnosti datoteka koje je možda dikirano i samim *Windows*-ima. Nakon toga, brzina prevođenja je veoma slična kod oba paketa, s tim što, zahvaljujući mogućnosti predkompajliranja zaglavlja, ceo proces kod složenih projekata može da bude znatno kraći sa „Borland" kompajlerom. S druge strane, QC/Win može dostići BC++ upotrebom opcije **incremental compile**, tako da ni jedan od dva paketa nema izrazitu prednost po pitanju brzine. Ono što je primetno je nešto čvršće vezivanje resursa računara od strane BC++ u toku prevođenja – postoje trenuci kada se veoma teško možete prebaciti u drugi *Windows* task. To se posebno odnosi na fazu povezivanja, kada vam se može desiti „totalno zamračenje" sisteme ako pokušate, dok traje prevođenje, da se iz DOS taska, u koji ste prešli, vratite nazad u BC++ okruženje, sve dok linker ne privede posao kraju.

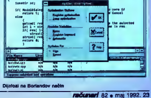
Prijavljivanje grešaka je takođe sasvim klasično. U odvojenom prozoru se pojavljuje spisak grešaka – postavljanjem kursora na neku od njih i pritiskom na Enter prelazimo u prozor u kome nas čeka datoteka i kursor na tačnom mestu greške – ne samo na pravoj liniji, već i unutar linije, na onom mestu koje je za kompajler sporno.

SPECIJALITETI KUĆE

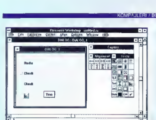
Ovim se ujedno završava lista opcija koje se standardno mogu naći kod oba okruženja i počinje lista onih opcija koje su ili specijalitet „Borland"-a ili su urađene na specifičan način. Prva među njima je **object browser** – opcija koja omogućava šetnju po simbolima (funkcijama i promenljivama) koje se koriste unutar izabrane klase. Možete početi od grafičke mape klasa prikazanih u njihovom uzajamnom odnosu – izborom neke od njih prelazite u novi pro-



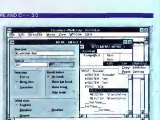
Integrisano okruženje



Dijalozi na Borlandov način



Pravi *Windows Resource* editor: kreiranje dijaloga...



... složenih menija ...

zor u kome je dat detaljan sadržaj klase. Sada vam je na raspolaganju pregled karakteristika izabranog elementa, prelaz u izvorni kod na mesto na kome se nalazi, vraćanja korak u nazad ili, ponovni skok na osnovnu mapu klasa.

Drugi element koji je specifičan je „Borland"-ova odluka da ne ugradi integrisani dibager u samo okruženje, već da se koristi *Windows* verzija *Turbo Debuger*-a.

Posledice ovakve odluke se brzo vide – startovanjem dibagera iz osnovnog menija prelazimo u tekst režim sa poznatom slikom i potpuno gubimo kontakt sa *Windows* okruženjem. Displej će se vraćati *Windows*-ima samo kada treba prikazati rezultat izvršavanja programa što je, ako program izvodite korak-po-korak, krajnje neprijatno zbog stalnog „bljeskanja" ekrana pri prelasku iz grafičkog u tekst režim i nazad. Od toga što se u *Windows* editoru i dalje nalazi vaš izvorni program nema nikakve koristi – rad sa dibagerom se odvija potpuno nezavisno, što znači da za izmenu eventualno primećenih grešaka treba potpuno napustiti seansu (iz dibagera se ne može preći u *Windows* na klasičan način!) i ručno pronaći mesto na kome smo u dibageru stali. Dibager ove snage sposoban da radi pod *Windows*-ima je zaista lepo imati – ipak, u ovom okruženju od daleko veće svakodnevne koristi bi bio integrisani dibager, makar najjednostavnijih mogućnosti.

WORKSHOP

Nezamenljiv alat kod razvoja *Windows* aplikacija je i editor resursa – menija, ikona, kursora itd. „Borland" je ovom problemu prišao daleko ozbiljnije od „Microsoft"-a, pa je jednim programom objedinio razvoj svih elemenata, tretirajući ih poput elemenata nekog projekta. WORKSHOP programom možete kreirati sve – od ikona, bit-mapa, kursora i fontova preko dijaloga i menija, do string tabela.

Dijalog editor je veoma blizak onome što smo videli kod „Microsoft"-a. Dizajn je, naravno, potpuno drugačiji, ali je moguć rad sa svim kontrolama, a uz to i sa nekoliko dodatnih „Borland"-ovih, kao što su stilski drugačiji **check boox** i **radiop button**. Kreiranje menija je izvedeno na jednostavan i efikasan način – kretanjem po strukturi menija ubacujete ili izbacujete pojedine stavke, uz istovremeni prikaz njenih karakteristika na jednom delu ekrana i efekta upravo napravljene izmene na končan izgled menija. Originalnosti „Borland"-u nije nedostajalo ni ovde – pored klasičnih kaskadnih menija, sada imamo mogućnosti postavljanja **menu break** na neku stavku menija – time se tekući meni prekida i ova opcija zajedno sa svim iza nje prelazi u novu kolonu. U njoj možete postavljati dalje prekide, ako da je ukupan efekat višekolonski meni sa ili bez horizontalnih linija razdvajanja. Osnovna razlika u odnosu na kaskadne menije je što se cela struktura pojavljuje odmah nakon što se u uđe u granu – dosta pregledno rešenje koje se može dobro iskoristi.

Najrazrađeniji je, svakako, modul za obradu bit-mapa i ikona. Crtanje ikona za *Windows* postaje pomalo i takmičarska disciplina među korisnicima, ali je sa dosadašnjim alatom taj posao bio prilično zamaran, jer je najveći broj tačaka na kraju ipak bilo potrebno pojedinačno „upaliti". Iako se njima prvenstveno kreiraju sličice veličine do 32×32 bita, „Borland"-ov grafički editor je urađen potpuno u maniru „velikih" editora. Bit-mapa nad kojima operiše tretira na isti način – sve jedno je da li editujete ikonu 32×32 tačke, ili bit-mapu od 32000×32000 tačaka. Opcija mu zaista ne nedostaje – od osnovnih za popunjavanje površina i crtanje geometrijskih figura pa do pisanja teksta i vazdušne četkice. Efekat primene ovih alata na ikonama je veoma interesantan – iako u zadatu površinu možete ubaciti svega nekoliko slova a svaki potez vazdušnom četkicom obično završi van ikone – dobijeni rezultat se ipak izdvaja originalnim izgledom.

WINDOWS ALATI

Od ostalih alata za *Windows* okruženja tu je pre svega *Profiler* ali je njegova *Windows* implementacija identična onoj koja je primenjena za dibager, pa zato i o njemu nekoliko reči kasnije kada bude reči o DOS okruženju. Pravi *Windows* alat je *WinSight* – više nego dobrodošla zamena za korisan ali skroman *Spy* iz Microsoft SDK paketa. Iako obavlja potpuno isti posao – praćenje promena poruka na sistemu – *WinSight* to čini daleko preglednije i informativnije od svog konkurenta. U dva odvojena prozora prikazani su hijerarhijska lista svih prozora na sistemu i poruke koje prolaze sistemom. Najveću vrednost ovom programu daje način na koji se prikazuje sadržaj poruka – ne samo da se vidi naziv poruke i pošiljalac, već se i ostali podaci prikazuju koliko je to moguće tekstualno, zajedno sa eventualnim rezultatom obrade poruke. Tako će, recimo, poruka WM_GETTEXT biti prikazana zajedno sa stringom koji funkcija za obradu vraća.

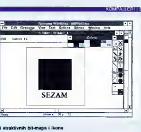
Još nekoliko manjih alata zaokružuju *Windows* verziju BC++ – IMPLIB za kreiranje *import* biblioteka, TDSETUP za konfigurisanje veze za korišćenje dibagera na udaljenom računaru, program za konverziju stringova iz ANSI u OEM i nazad, „Misrosoft" kompatibilan *Resource Compiler* i *Help Compiler* za generisanje standardnih *Windows* help datoteka.

Konačno, uz *Windows* okruženja ide i oko 4MB help datoteka kojima je pokriveno sve što se u ovom okruženju koristi – od editora i ostalih alata, preko C++ jezika, do samog *Win API*-a. Help se bazira na standardnom *Windows* help mehanizmu, a sadržaj je izuzetno obiman – bolji referentni priručnik od njega se teško može zamisliti.

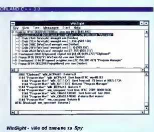
Ukupan utisak o ovom delu paketa BC++ 3.0 je pomešan. Paket je zaista kompletan, neki njegovi delovi su do sada najbolje što je od takvih programa za *Windows* okruženja viđeno, ali sa druge strane ostaje određena nedorečenost kod dibagera i profajlera koji, iako rade sve što treba, ipak ne spadaju u pravo *Windows* okruženje. Da je jedina posledica primenjenog „DOS-look" rešenja problem bi se mogao svrstati među estetske, pa čak i rešenje braniti navikama korisnika, ali na žalost nije – operativnost je znatno umanjena. Nisam imao tehničkih mogućnosti da probam upravljanje seansom sa udaljenog računara – moguće je da bi u takvim uslovima rešenje izgledalo bolje.

DOS OKRUŽENJE

Ako je *Windows* okruženja krije dosta novosti, za DOS integrisano razvojno okruženje se može reći da sledi tradiciju „Borland"-ovih ranijih proizvoda. Uostalom, ovaj koncept i jeste na neki način „Borland"-ov patent, pa nije iznenađenja da je njegova realizacija možda i najbolja u ovom trenutku. Za razliku od „Microsoft"-ovih, rekao bih gotovo nespretnih poteza sa glomaznim PWB-om, BC++ 3.0 nudi takođe glomazno ali daleko brže i operativnije rešenje. Startovanjem BCC programa dobijamo poznat izgled ekrana na kome se, sredstvima dostupnim u tekst režimu, ostvaruje utisak *Windows* okruženja: tu je više prozora sa kontrolama za zumiranje na punu dimenziju ekrana, svođenje na manji format, zatvaranje prozora, pa čak i na nešto što se može smatrati ikonom. Odziv sistema na sve komande je trenutan, pa čak ni privremeni izlazak u DOS koji zahteva prebacivanje dela memorije na disk kako bi se DOS-u ostavilo što više memorije traje zanemarljivo kratko. Brzina je oduvek bila jedna od karakteristika koje je „Borland" stav-



... i atraktivnih bit-mapa i ikona



WinSight - više od zamene za *Spy*

ljao na vrh liste – ovo okruženje u svemu održava tradiciju

Raspored opcija po menijima i način rada je identičan onome pod *Windows*-ima – od otvaranja projekta preko podešavanja niza opcija do generisanja EXE programa ugrađenim make modulom. Za razliku od *Windows* verzije, odavde je moguće generisati ne samo DOS EXE već i *Windows* EXE i DLL formate, ali je svakako najvažnija razlika postojanje pravog integrisanog dibagera. Njegove mogućnosti su, naravno, sasvim drugačije koncipirane od *Turbo Debug*-era, ali mu je i namena drugačija – brzo pronalaženje grešaka isključivo u izvornom kodu.

Skup alata za DOS okruženje, pored integrisanog razvojnog okruženja čine i odvojen kompajler, *Turbo Assembler*, *Turbo Debuger*, *Turbo Profiler*, *Turbo Linker* i *Make*. Karakteristika svih ovih programa je da uključuju DPMI (*DOS Protected Mode Interface*) koji im omogućava rad sa svom raspoloživom memorijom u računaru. To konkretno znači da je kompajler u mom računaru bio u stanju da iskoristi svih 4 MB memorije, optimizujući tako proces prevođenja smanjenjem potrebe za korišćenjem diska.

C++ KOMPAJLER

BC++ 3.0 je kompajler namenjen prevođenju C i C++ programa prema AT&T specifikaciji v.2.1, odnosno sa punom ANSI C kompatibilnošću. Namera proizvođača je da ovaj kompajler izdvoje od ostalih pre svega po dve karakteristike – brzini prevođenja i optimizaciji koda. Od optimizacionih tehnika primenjene su one koje srećemo i kod ostalih kompajlera, kao što su globalna alokacija registara, izbacivanje „mrtvog" koda, izbacivanje višestrukih ponavljanja istog koda, vađenje konstantnih delova iz petlji, pamćenje izračunate vrednosti izraza i direktna upotreba vrednosti umesto ponavljanja izračunavanja izraza, uvođenje dodatnih promenljivih za pamćenje međuvrednosti kako bi se pojednostavilo izračunavanje izraza itd. Sve ove tehnike srećemo i kod drugih kompajlera, ali je primenjeni algoritam taj koji odlučuje da li će optimizacija stvarno imati značajnijeg efekta. Bar po dosadašnjim verzijama, „Borland" se optimizacijom nije proslavio – razlike između neoptimizovanog i maksimalno optimizovanog koda su bile minimalne. Provera kvaliteta i pouzdanosti optimizacije, na žalost, zahteva dosta vremena – u ovom trenutku još nemam rezultata na osnovu kojih bi se mogao izvući neki pouzdan zaključak.

Ako je već provera kvaliteta generisanog koda problem, onda provera brzine prevođenja sigurno nije – BC++ 3.0 je zaista izuzetno brz kompajler koji je dosta odmakao „Microsoft"-u. Kao što smo već spomenuli kod *Windows* verzije, pretkompajliranje zaglavlja je jak dodatni adut. „Borland" je napravio još jednu novinu kojom pokušava da prevaziđe problem DOS memorije i veličine programa. Standardne DOS **Linker Overlay** tehnike nisu nikada stekle neku posebnu popularnost. Zato je razrađen sopstveni sistem nazvan VROOMM – *Virtual Run-time Object Oriented Memory Manager*, koji skoro potpuno transparentno za programera omogućava generisanje neograničeno velikih programa.

Interesantno je da, i pored sve inventivnosti i želje da bude prvi, „Borland" priznaje da je „Microsoft" taj koji nameće standarde. Kompatibilnost ove verzije C kompajlera sa MSC 6.0 je skoro potpuna – razlika je jedino u i onako malo korišćenim **based pointer**-ima, koji su „Microsoft"-ov izum. „Borland" je napravio i niz makroa kojima usaglašava nazive nekih nestandardnih funkcija, a svi se mogu aktivirati definisanjem __MSC makroa, nakon čega BC nudi punu MSC kompatibilnost.

DOS ALATI

Turbo Debuger 3.0 je dosledni nastavak ranijih verzija sa vrlo malo novosti, ukoliko se naravno izuzme verzija za *Windows*. Glavna novost je mogućnost korišćenja lokalne mreže za udaljeno upravljanje dibagerom, koja je takođe primenjena i kod *Profiler* programa (u paketu je verzija 2.0). U ovom drugom se krije i više noviteta – umesto interaktivnog rada i analize sada je na raspolaganju i **batch** režim rada – rezultati analize se skupljaju u posebnu TFS datoteku i naknadno analiziraju. *Turbo Profiler* predstavlja izuzetno vredan alat za analizu programa – osnovni zadatak mu je da prati tok izvršavanja programa i meri parametre – provedeno vreme u pojedinim delovima koda, broj pozivanja potprograma i slično. Pored toga, *Turbo Profiler* je u stanju da prati i sve aktivnosti nad datotekama ali i interapt pozive. Kada se ovome doda i nova mogućnost detektovanja delova koda koji nikada nisu izvršeni u toku rada programa, dobija se zaista izuzetan alat za optimizaciju programa.

Kao i ostali „Borland"-ovi proizvodi, i *Turbo Assembler* je stekao reputaciju najbržeg prevodioca. Uz tu reputaciju obično idu blago nestandardna rešenja što je i ovde slučaj. Turbo Assembler je zaista „napakovan" karakteristikama – od začetaka objektnog programiranja, do posebnog režima nazvanog IDEAL u kome je sintaksa nešto izmenjena i prilagođena onome na šta smo navikli kod viših programskih jezika. I ovde je, slično kao kod C kompajlera, omogućena MASM kompatibilnost prelaskom u MASM režim. Sve to, naravno, uz mogućnost generisanja punog 386 ili 486 mašinskog koda.

TRKA SE NASTAVLJA

Zaista je teško, čak i u tekstu malo većeg obima makar i pomenuti sve što se nalazi u ovom paketu. Prvi, opšti utisci su da je njegova vrednost pre svega u kompletnosti i proverenom kvalitetu delova namenjenih DOS okruženju. *Windows* verzija još uvek ne deluje razrađeno i usaglašeno do kraja, što odaje možda i žurbu da se na izazov reaguje bez zaista kompletnog rešenja. I dokumentacija odaje takav utisak – sve što se tiče *Windows* okruženja je samo „umešano" u DOS uputstva, što se eventualno može braniti idejom da se prosto radi o istim proizvodima koji se samo neznatno razlikuju od okruženja do okruženja. Ipak, verujem da će u sledećim verzijama biti napravljena jasnije razlika, koju ova „dva sveta za sebe" objektivno i nose. Uostalom, i alati koje „Borland" pakuje u ovu verziju pokazuju koliko se ta dva sveta razlikuju.

BC++ 3.0 u ovom trenutku predstavlja paket izuzetne vrednosti, kome nema konkurencije. „Microsoft" je već najavio svoj odgovor – MSC 7.0 koji ćemo takođe prikazati čim ga dobijemo na test. Ako je suditi po dosadašnjoj praksi, može se opet očekivati uspostavljanje ravnoteže koju će neka od ove dve firme morati da pokuša da promeni u svoju korist. Takva situacija je svakako najbolje moguća za nas – svako može ostati na onom proizvodu koji je ranije izabrao, ne menjajući tako navike, a novi korisnici neće napraviti grešku za koji god proizvod da se opredele.

BC++ 3.0 ćemo se sigurno još koji put vraćati na stranicama „Računara". Verujem da će rezultati njegove upotrebe u svakodnevnom poslu zahtevati da se o njegovoj pouzdanosti, brzini i kvalitetu generisanog koda napiše još nešto.

Korisna adresa

SOFTLAND
11000 Beograd, Skadarska 45
Tel. (011) 848-5698

BISER IZ SENKE

Najnoviji „Amtecov" proizvod za grafički XY, 2D/3D prikaz rezultata inženjerskih proračuna omogućuje jednostavno kreiranje i manipulisanje podacima i visokokvalitetne kopije crteža na skoro svim ploterima i laserskim printerima.

Saša Vlajić

Pomalo u drugom planu, u sjeni CAD grafičkih programa, egzistira klasa grafičkog softvera namijenjena užem krugu korisnika, naučnicima i inženjerima, o kojoj se rjeđe piše. To ni u kojoj mjeri ne umanjuje njen značaj i potražnju. Radi se o programima za vizuelni prikaz naučnih i inženjerskih podataka tzv. „žičanom mrežom" u 2D i 3D, krivim linijama u XY ravni, graficima itd.

Evo nas u prilici da upoznamo jednu zanimljivu, kvalitetnu kreaciju sa ove scene, program *Tecplot*. U pitanju je njegova peta verzija, rezultat više od deset godina iskustva renomirane kuće „Amtec".

NA ULAZU...

Hajde da prvo navedemo ono što čini, kako bi matematičari rekli, potreban uslov da bi nam *Tecplot 5.0* pokazao šta zna. Za njegovu PC verziju minimum je 286 mašina sa 4 MB RAM-a. Numerički koprocesor učiniće da rad bude efikasniji i ugodniji, a što se grafičke karte tiče, program nije izbirljiv – u SETUP-u treba markirati jednu od 157 ponuđenih (mono ili kolor). Svakako, tu je još i 649 USD – za svijet van ove naše balkanske konde umjerena cijena softvera takvih performansi kakve *Tecplot* posjeduje.

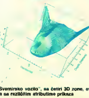
Informacije radi, postoje i verzije programa za skoro sve tipove radnih stanica: „Sun" „SPARCstation", SGI „Iris" i „Indigo", IBM RS6000, HP/"Apollo" Unix i Domain, DEC, VAX, MIPS," Data General", itd.

Tecplot 5.0 je zapravo integrisani program za crtanje XY, 2D, 3D kontura, silnica (strujnica) vektorskih polja, vektora i svega onoga što je potrebno da bi se prikazali rezultati proračuna u, na primjer, takvim oblastima kao što su teorija elektromagnetnog polja, termodinamika, dinamika fluida i slično.

Ulazni podaci moraju biti u formi standardnog ASCII fajla, bilo da je on kreiran teksteditorom, uvezen iz *spreadsheet* (na primjer, *Lotus 1-2-3*) ili nekog specijalnog programa. Taj ulazni fajl sastoji se od zaglavlja (nije obavezno) sa do tri linije koje definišu ime fajla, imena promjenljivih i broj tačaka kojima se pridružuju podaci. Nakon toga slijedi jedna ili više grupa podataka, specificirajući tzv. zone. Svaki prikaz, crtež, može biti podijeljen na do 2048 dijelova, zona, omogućujući nam nezavisan pristup svakoj od njih. Tako možemo izabrati različite atribute po zonama i u jednoj uzeti prikaz u obliku mreže, u drugoj u obliku osjenčene površine, itd. Dakle, što imamo više zona, ulazni fajl će sadržati više grupa podataka, biće složeniji.

Sama grupa podataka daje se u vidu izraza nalik na fortranske (s obzirom na najveću zastupljenost fortrana među naučnicima i inženjerima). Skaliranje, inverzija (mirorovanje) i rotacije definišu se vrlo lako. Na raspolaganju su matematičke funkcije: SIN, COS, TAN, ASIN, ACOS, SQRT, ABS, ALOG, ALOG10 i EXP.

Da ne previdimo, postoji i mogućnost obrnutog redosljeda kod definisanja ulaznog fajla. Korištenjem miša može se nacrtati multisegmentna linija u 2D/3D domenu, kako bi *Tecplot* izvršio interpolaciju kubnim splajnom u njenim



„Svemirsko vozilo", sa četiri 3D zone, svaka sa različitim atributima prikaza

tačkama. Interpolirane vrijednosti se spremaju kao ulazni fajl, tako da se naknadno može izvršiti učitavanje i crtanje XY prikaza.

Numerički podaci koji se odnose na zonu mogu biti dati u **POINT** i **BLOCK** formatu. U prvom slučaju se grupišu vrijednosti svih promjenljivih za prvu tačku prikaza, zatim za drugu, i tako redom. U **BLOCK** formatu grupišu se vrijednosti prve promjenljive za sve tačke, zatim druge promjenljive za sve tačke itd.

RADNA OKOLINA

Ona je standardna, sa padajućim menijima i stupcem za razne podatke (tekuće X,Y,Z koordinate, uglovi, izbor zone,...) sa lijeve strane velikog pravougaonika u centru – površine za crtanje.

Paleta prikaza koji se mogu dati na tom prostoru, kao što je već u uvodu rečeno, izuzetno je široka: Kontura, mreže, vektori, grafici sa stupcima (*bar charts*), istačkani crteži (skice),... Za ilustraciju, 3D prikaz, sa svim zonama od kojih se sastoji, možemo osmotriti iz bilo kog ugla, osvijetljenog difuzionom svjetlošću proizvoljne boje. Izvor svjetlosti mišom fiksiramo po želji.

Veoma elegantno je riješen rad sa prozorima. U jednom momentu može ih biti do šesnaest, različitih veličina i prikaza koje sadrže. Preklapaju se ili su jedan u drugom, postoji mogućnost izbora boje pozadine i okvira. Prave vratolomije mogu se izvoditi rotacijom oko bilo koje koordinatne ose ili ose okomite na površinu ekrana, nezavisnom izmjenom razmjere svake od njih, miksanjem zona, zumiranjem.

Zanimljiva alatka je „sonda" (zapravo, kurzor miša u obliku krstića), nazvana tako vjerovatno radi analogije sa sondom osciloskopa. Njenim postavljanjem na bilo koju tačku već gotovog crteža vrši se interpolacija vrijednosti promjenljivih na osnovu podataka za susjedne tačke. U stanju smo, dakle, očitati vrijednosti svih promjenljivih na bilo kojem dijelu crteža.

Inače, jednu tačku u ulaznom fajlu može definisati do 2048 promjenljivih. Broj tačaka ograničen je raspoloživom *extended* memorijom (VMS i UNIX radne stanice praktično ne poznaju ovaj problem).

Kreatori *Tecplot*-a nisu zaboravili ni upisivanje teksta. Mogućnosti su i ovdje, kao kod samih crteža, mnogobrojne: izbor veličine, ugla ispisa, boje, fonta, pozadine, poravnavanja... Od fontova spomenimo samo **Helvetica/Bold, Greek, Math, Times/Normal/Italica/Bold, Courier/Bold**.

MAKROI

Jednu od ponajboljih strana programa *Tecplot 5.0* čine makroi. Svi znamo kolika je olakšica eliminisanje potrebe da se neke monotone operacije ponavljaju n puta, n beskonačno. Kod programa ovog tipa, gdje se rukuje mnoštvom detalja, makroi su od izuzetnog značaja. Ovdje ih je zaista jednostavno kreirati: skoro četrdeset makro-komandi, plus mogućnost kreiranja makro-fajlova ASCII tekst-editorom. Postati dobar makro-bibliotekar sada je stvarno lako.

Operacije nad podacima izvršavaju se pozivanjem eksternih programa pisanih u C-u iz *Tecplot*-ovog „Data menija". Dva dolaze uz originalni program (*Preplot* i *Framer*; na primjer, *Framer* služi za plejbek bit-mapiranih prikaza na ekranu – jednostavan i jeftin način animacije), a u dokumentaciji je podrobno opisano kako se mogu načiniti vlastiti.

...I NA IZLAZU

Kada govorimo o izlazu iz algoritma zvanog *Tecplot 5.0*, odnosno dobijanju crteža na papiru, nepotrebno je napominjati da ovakav program podržava sve tipove najkvalitetnijih plotera i printera. Podržani su formati: HPGL, HPGL2, *PostScript*, *Color PostScript*, enkapsulirani *PostScript* (mogućnost povezivanja sa programima za stono izdavaštvo), *Lasergraphics*, i drugi.

Sjetimo li se, na kraju, poslovice koja kaže da je najskuplja neiskorištena stvar, onda će ulaganje u ovaj softver stvarno biti rasipanje novca za one sa skromnijim potrebama (što nikako ne podrazumijeva i skromnije znanje). Za pola cijene mogu se naći pristojni programi iz niže klase (na primjer, *EasyPlot 2.2* – 349 USD). Onima koji ga nabave, *Tecplot* će sigurno pružiti mnogo zadovoljstva tokom rada.

Lična karta

Program: *Tecplot 5.0* (15.1.1992.)
Namjena: prikaz naučnih i inženjerskih podataka
Sadržaj paketa: dvije kopije softvera (jedna na četiri 5.25" HD diskete, a druga na tri 3.5" DD) zajedno sa eksternim programima (*Preplot*, *Framer*), dokumentacija (Uputstvo za instalaciju, „User's Manual" na preko 300 stranica, te odličan „Quick Reference Guide"), registarska kartica
Cijena: 649 USD u SAD. EUSS (Extended Update and Support Service): 129 USD godišnje
Potrebno hardversko-softversko okruženje: IBM AT/386 ili kompatibilan, 4 MB RAM-a, hard disk, tekst-procesor
Prostor na disku: nešto manje od 5 MB
Ulaz: tastatura, miš
Izlaz: ekran – bilo koja grafička karta, gotovo sve vrste plotera i printera
Proizvođač: AMTEC Engineering Inc., P.O.Box 3633, Bellevue, WA 98009-3633, tel. 991/1-800-676-7568

STARI PRIJATELJ U NOVOM RUHU

Davna 1954. godina smatralo se da je najveći problem u programiranju računanje matematičkih formula. Naučnici u IBM korporaciji uamarili su napore na stvaranje programskog jezika koji će moći da prevodi formule, pa je tako prvi viši programski jezik u istoriji računarstva dobio ime FORTRAN, FORmula TRANslator, prevodilac formula. Fortran je ubrzo postao *lingua franca* računarskog sveta: 80% svih računara bilo je isporučivano sa odgovarajućim prevodiocem. Progrsmiranje na fortranu i danas predstavlja veliki biznis. „Microsoft" uživa da u svakoj čorbi bude mirođija, pa predstavljamo najnoviju verziju MS FORTRAN-a, sa rednim brojem 5.1.

Duško Savić

Fortran su pravili praktičari za praktičare, a pedesetih godina jedino što se moglo raditi sa računarima je da se programiraju. Naučnici svih tehničkih profila prigrlili su fortran i napisali stotine hiljada programa među kojima postoje čak i programi od po milion programskih redova. Takvu programsku bazu je nemoguće ignorisati, pa je sastavni deo novih verzija fortrana kompatibilnost sa prethodnim verzijama. Broj dijalekta fortrana ubrzo je postao prevelik i 1966. godine izvršena je prva velika standardizacija jezika. Sledeća je bila 1977. godine a poslednja 1990. Zato se govori o kompatibilnosti sa fortranom 66, 77 i 90. Osim ovih važna je i praktična kompatibilnost sa verzijama na nekad popularnom hardveru IBM 360 i 370, odnosno VAX. MS FORTRAN 5.1 na sve to dodaje i sopstvena proširenja koja su u skladu sa prirodom modernih PC računara: DOS, *Windows*, OS/2.

Fortran se tokom decenija menjao (mada znatno manje nego što bi se to moglo pretpostaviti ili želeti) a i fortran programi očevidno nisu u većini. „Naučno programiranje" (*scientific programming*) je danas rezervat u svetu prepunom C, Pascal i dBASE programera, ali je vrednost fortranskih programa i dalje velika. Ako program treba da sadrži iole komplikovanije matematičke funkcije, jedini pravi jezik je bio i ostao – fortran.

PAKET I INSTALACIJA

Priručnici su *Reference* (534 strane), *Quick Reference Guide* (132), *Installing and Using the Professional Development System* (86), *Advanced Topics* (362) i *Environment and Tools* (666). Program se isporučuje kao niz kompresovanih datoteka na ukupno šest disketa formata 5.25 inča i kapaciteta 1.2 megabajta. *Reference* definiše jezik i proces prevođenja i povezivanja (*linking*). Štampan je u dve boje: crnim slovima se opisuje standardni fortran, a plavim – Majkrosoftova proširenja. Skoro pola teksta je u plavim slovima.

Quick Reference Guide je uvezan plastičnim spiralama i očigledno je namenjen svakodnevnom prelistavanju i korišćenju. U njemu su nabrojane sve opcije za prevodilac koji se aktivira iz DOS-a (program FL), dat je prikaz svih naredbi jezika, ugrađenih funkcija i dodatnih procedura, grafičkih rutina, metanaredbi za fortran, opcije u isporučenom programskom editoru, naredbi za dibager *CodeView*. Objašnjeni su i pomoćni programi: BIND (konvertuje OS/2 program tako da se može izvršavati i pod DOS-om i pod OS/2), CVPACK (smanjuje veličinu datoteke koja je bila pripremljena za *CodeView*), EXEHDR (menja zaglavlje EXE ili DLL datoteke), EXP (briše imenike koje editor pravi tokom rada), HELPMAKE (program za stvaranje help datoteka), ILINK (obnovljeno inkrementalno povezivanje projekta), IMPLIB (uvozne biblioteke za DLL datoteke), LIB (stvaranje i održavanje biblioteka), LINK (povezivanje datoteka u EXE ili DLL datoteke), NMAKE (automatizacija projekata), PWBRMAKE (interna konverzija datoteka za PWB), QuickHelp (prikazivanje pomoćnih datoteka), RM (briše skrivene imenike koje kao pomoćne pravi PWB), UNDEL (suprotno od RM).

Priručnik *Environment and Tools* detaljno objašnjava rad sa isporučenim editorom po imenu *Programmer's Workbench*, a slede uputstva za *CodeView*, LINK, ILINK, NMAKE i druge pomoćne programe. Priručnik *Advanced Topics* bavi se opširnijim programima, strukturom datoteka, brojevima u pokretnom zarezu, multimodularnim i višejezičkim programiranjem, grafikom i posebnostima programiranja za *Windows* i OS/2.

Instalacija se izvodi programom SETUP. Tokom instalacije treba se odlučiti za operativni sistem, vrstu biblioteke sa kojom će prevodilac raditi, C kompatibilnost, način izračunavanja brojeva u pokretnom zarezu, memorijske modele, vrstu grafike, instaliranje isporučenog editora PWB, BRIEF kompatibilnost, drajver za miša, programske primere, pomoćne datoteke, imenike i diskove na kojima će se nalaziti instalirani FORTRAN.

Dve su vrste operativnog sistema za koje se treba odlučiti: sistem pod kojim će se praviti program (DOS i OS/2) i sistem za koji se program pravi (DOS, OS/2 i *Windows*). Sve kombinacije ipak nisu dozvoljene: programi pravljeni pod OS/2 moraju se izvršavati pod OS/2, mada je moguće praviti i OS/2 programe koji se izvršavaju i pod DOS-om. Za *Windows* programe nema smetnje: prave se pod DOS-om, a izvršavaju pod *Windows*-ima.

Za svaku kombinaciju memorijskog modela, načina rada sa brojevima u pokretnom zarezu i ciljnog operativnog sistema potrebno je napraviti posebnu vrstu biblioteke. SETUP omogućava da se napravi bilo moguća kombinacija ali istovremeno i pita da li treba da se unište originalne biblioteke (biblioteke iz kojih se izvode razne kombinacije) da se ne bi zauzelo previše mesta na disku.

Posebna opcija je priključivanje grafičke biblioteke GRAPHICS.LIB, koja ima smisla samo pod DOS-om.

Za rad sa brojevima u pokretnom zarezu postoje tri mogućnosti: softverska emulacija (koprocesor se koristi ako je prisutan, ali program radi i ako koprocesora nema), direktno korišćenje koprocesora (program ne radi ako koprocesora nema) i alternativna matematička biblioteka, koja nikada ne koristi koprocesor (čak i ako je ovaj prisutan).

Kraj rada programa SETUP ne znači i da je instalacija izvršena do kraja. SETUP stvara pomoćne datoteke sa izmenama koje korisnik treba ručno da unese u datoteke CONFIG.SYS i AUTOEXEC.BAT. Ako se taj dodatni korak ne izvrši, prevodilac neće raditi. SETUP stvara i datoteku TOOLS.PRE. U njoj su razni ulazni podaci za editor PWB, s tim što PWB očekuje da se ta datoteka zove TOOLS.INI. Ako ništa drugo, treba preimenovati TOOLS.PRE u TOOLS.INI i to je upravo ono što će korisnik koji se prvi put susreće sa MS FORTRAN-om i uraditi. Slično, ako se kao ciljni operativni sistem odabere *Windows*, sadržaj datoteke TOOLS treba ubaciti u WIN.INI. Sve u svemu, trebalo je i ovaj zadnji korak sistematizovati, a ne ostaviti korisniku da frustrirano manuelno iskušava razne varijante. Primera radi, meni je instalacija uspela tek iz trećeg puta, i to pošto sam naznačio da sve datoteke treba smestiti u jedan jedini imenik, po imenu FORTRAN. Posle prve instalacije, na primer, SETUP je javio da je sve proteklo u redu, jedino što nijedan program nije mogao da se prevede. U drugoj instalaciji sam dodao i *Windows* kao ciljno okruženje i SETUP je samo napravio još nekoliko imenika više po dubini, i opet ništa nije radilo. Ali treća – sreća. Instalirao sam FORTRAN samo za DOS i sve u jedan imenik, i onda je sve proradilo.

Instalirani fortranski sistem zauzima 5-6 megabajta na disku.

MEMORIJSKI MODELI

Ključno otkriće u razvoju računarstva predstavlja je fon Nojmanove ideje da se programi i podaci drže u memoriji istovremeno. I danas svaki računar radi upravo tako, s tim da program sam zna gde su podaci. U PC računarima memorija nije linearna nego je podeljena na segmente dužine 64K. To je veliko praktično ograničenje, jer ni program ni podaci ne mogu biti duži od 64K. Uz izvesnu dozu lukavosti sva ta ograničenja padaju, ali se onda prevodilac i programer bave poslom kojim ne bi trebalo da se bave. Zato jezici na PC računarima definišu memorijske modele – konvencije o prevazilaženju ograničenja prostenih je prisustva segmenata. Tako MS FORTRAN podržava tri standardne memorijske modela: srednji (medium), veliki (large) i ogromni (huge). Oni se definišu ovako:

vrsta modela	ograničenje dužine naredbi	ograničenje dužine podataka
srednji	nema	64K
veliki	nema	nema
ogromni	nema	nema

Osnovni model je **large**. U njemu ukupna dužina naredbi i podataka svaka za sebe može prevazići dužinu segmenta. Međutim, za svaki modul određuje se po jedan segment, tako da su moduli ograničeni na 64K. Formalni argumenti nizova ograničeni su na 64K (osim ako nisu deklarisani sa HUGE), a prevodilac sam stvara višestruke segmente naredbi i podataka, po potrebi. Model medium ograničava dužinu programa i pojedinačnih modula na 64K, s tim da ukupna dužina programa može prevazići 64K. Model **huge** je sličan modelu **large**, osim što se podrazumeva da svi formalni parametri nizova prevazilaze 64K.

Zajedničko za sve modele je to što postoji osnovni segment podataka, i što je on – uprkos svoje važnosti – ograničen na 64K. U njemu se nalaze stek (sadrži argumente koji se prenose potprogramima), konstante tipa REAL i CHARACTER, podaci koje koriste izvršne biblioteke, razni baferi, nizovi koji ne prelaze neku granicu, ulazne adrese za potprograme i sl. U medium modelu tu se nalaze i COMMON blokovi (koji nisu deklarisani kao HUGE ili FAR). U većim programima 64K za sve to postaje strašno malo, pa postoje određeni postupci da se podaci iz osnovnog segmenta rasporede po drugim segmentima.

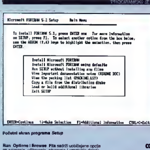
Lepo je što sve to može, ali nije lepo što to programer mora eksplicitno da zahteva.

Fiksni nizovi duži od 64K automatski se raspoređuju u onoliko segmenata koliko im je potrebno, sve do krajnjih granica raspoložive memorije. (Prevodilac takve nizove automatski deklariše kao HUGE.) Međutim, nijedan pojedinačni objekat ne sme preći granicu od 64K, npr. član niza većeg od 64K ne sme sam za sebe biti veći od 64K. Ako se radi o nizovima promenljive dužine, najbolje je deklarisati ih kao HUGE, da bi se predupredili problemi neusaglašenih dužina.

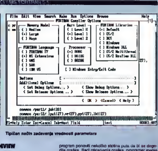
Naravno, sve se ovo odnosi na DOS. Ako se fortranski program prevede tako da radi pod *Windows*-ima, veličina niza postaje beskonačna, tj. do 16 megabajta ili čak i više u 386 režimu, naravno, pod pretpostavkom da toliko memorije stvarno ima u sistemu.

PROGRAMMER'S WORKBENCH – EDITOR

Sa nekoliko godina zakašnjenja, „Microsoft" je prihvatio „Borland"-ovu koncepciju integrisanog editora i sistema projekata iz kojih se aplikacije pišu direktno, a ne razdvojene po etapama ciklusa unos programa – prevođenje – povezivanje – izvršenje programa – dibaging, pa opet sve u krug. Isporučeni editor zove se Programmer's Workbench, skraćeno PWB. Opcije glavnog menija su: **File**, **Edit**, **View**, **Search**, **Make**



Početni ekran programa *Setup*



Tipičan način zadavanja vrednosti parametara

Run, **Options** i **Browse**. **File** sadrži uobičajene opcije za snimanje, učitavanje, otvaranje i umetanje datoteka. **Edit** sadrži opcije **Undo**, **Redo** i **Repeat**, koje su zaštitni znak svakog programerskog editora. Tu su i opcije **Cut**, **Paste**, **Copy** i druge, kao i snimanje i aktiviranje makronaredbi.

Opcija glavnog menija **View** otvara prozore na razne načine, a opcija **Search** pretražuje i/ili menja tekst. Moguće je obeležiti tekst na razne načine i onda direktno premeštati kursor na odabrane tačke. Moguće je preći direktno na greške nastale tokom prevođenja.

Opcija **Make** bavi se stvaranjem programa i predstavlja civilizovani način da se opšti sa programom NMAKE, koji usaglašava verzije pomoćnih datoteka koje nastaju tokom formiranja programa. Tu su i subopcije **Set Program List**, **Edit Program List** i **Clear Program List**, pri čemu se pod imenom **List** u PWB-u podrazumeva isto što **Project** u „Borland"-ovim editorima. Rezultat ovih opcija je tzv. programska lista, a to je zasebna datoteka naredbi koje treba izvršiti da bi se od teksta programa stiglo od izvršne verzije. Programska lista je analogon BAT datoteke u kojoj bi se našli uzastopni pozivi raznih delova fortranskog sistema. Za programe koji se sastoje iz jednog modula, npr. samo iz glavnog programa, programska lista je jednostavna, i skoro nepotrebna. Međutim, za programe sastavljene iz više modula (a svi veći fortranski programi moraju biti razdeljeni na module zbog ograničenja dužine modula na 64K), a pogotovu za programe sastavljene iz modula na različitim jezicima (C, paskal...) programska lista postaje dragocena.

Tehnički, NMAKE učitava sadržaj programske liste, dodaje informacije iz sistemskih varijabli i pogodnim redosledom poziva programe za prevođenje, povezivanje i prelistavanje datoteka. Jedan se može prevesti pod jednim uslovima, neki drugi modul pod drugim uslovima itd. Ako koristi ceo sistem onako kako je namenjen, mala je verovatnoća da će korisnik ikada morati da se bavi tehničkim detaljima programa NMAKE.

Opcije **Run** i **Options** glavnog menija sadrže očekivane subopcije za izvršenje programa, odnosno, fino podešavanje raznih delova sistema (editor, NMAKE liste itd.). Submeni za **Run** može se proširiti sa do šest novih programa. To mogu biti programi iz fortrana, uslužni programi (procesor teksta) itd.

Iz opcije **Browse** se može pronaći definicija bilo kojeg programskog simbola u tekstu programa i modula koji učestvuju u programu, prikazati pozivanje potprograma, videti koji potprogrami zovu druge potprograme, prikazati „kostur" programa, pronaći sledeće ili prethodno pojavljivanje imena i definisati podelu prozora u editoru.

Iz menija **Help** prikazuju se obaveštenja o pojedinim delovima celog sistema. Najjednostavnije upotrebe je da kursorom dođe na ime o kojem tražimo dodatne informacije i zatraži pomoć. Time se listanje obimne dokumentacije smanjuje na najmanju moguću meru.

CODEVIEW

U meniju **Run** naredbom **Debug** ulazi se u istraživanje programa dibagerom *CodeView*. Ovaj program se primenjuje na celu porodicu „Microsoft" jezika: asembler, jezik C, fortran i paskal. Da bi se *CodeView* mogao primeniti, prethodno treba ubaciti potrebne informacije u EXE program. I to se radi na poseban način u svakom od navedenih jezika. Za FORTRAN treba aktivirati opciju /Z, a zgodno je to što se ta opcija može naizmenično uključivati i isključivati, tako možemo posmatrati rad samo pojedinih delova programa.

Postoji čak 18 opcija za početak rada *CodeView*-a, a važnije među njima su korišćenje dva monitora istovremeno, izbor između *extended* i *expanded* memorije, odnosno diska, rad sa DLL datotekama itd. Trajna konfiguracija može se postići promenama datoteke TOOLS.INI. U datoteci CURRENT.STS *CodeView* pamti zaustavne tačke (*breakpoints*), izraza koje treba posmatrati tokom izvršenja programa, vrste prozora i razne opcije za svaki prozor ponaosob.

CodeView deli ekran na tri dela. U gornjem je glavni meni, u srednjem su prozori, a u donjem je statusna linija. Meni i status zauzimaju po red na ekranu, a srednji deo može biti razdeljen na više prozora, zavisno od preduzetih akcija.

Glavni meni sastoji se od opcija **File**, **Edit**, **View**, **Search**, **Run**, **Watch**, **Options**, **Calls** i **Help**. **File** služi za učitavanje teksta programa, odnosno modula, kao i za štampanje i odlazak u DOS. Subopcije **Undo**, **Copy** i **Paste** opcije glavnog menija Edit služe za prebacivanje teksta između prozora. Meni **View** sadrži naredbe za otvaranje, zatvaranje i promenu veličine raznih vrsta prozora, kao i za razgledanje rezultata programa na ekranu. Mogu se otvoriti prozori za tekst programa, stanje u memoriji, registrima, koprocesoru ili za help. Postoje i prozori za naredbe, razgledanje stanja lokalnih promenljivih i vrednosti izraza.

Opcijom **Search** pretražuje se po tekstu programa, ili po imenima funkcija i potprograma. U pretraživanju se mogu koristiti regularni izrazi. Na primer, moguće je naći sva legalna imena u C programu ili sve deklaracije INTEGER*2 u fortranu i sl.

Opcija **Run** sadrži tri grupe naredbi. Prva grupa sadrži naredbe za ponovno izvršenje programa, usporeno izvršenje programa i promenu ulaznih podataka. Druga grupa naredbi upravlja sposobnostima dinamičkog ponavljanja programa. Treća grupa naredbi se odnosi na rad pod OS/2 i omogućava razgledanje, manipulisanje i promenu programskih niti (*threads*) i procesa.

Dinamičko ponavljanje programa je važno iz više razloga. Omogućava da se lako ponovi kompletna prethodna seansa sa dibagerom, tako da novo istraživanje programa ne počinje uvek iz početka. (Naravno, što se više program promeni, to je dinamičko ponavljanje kroz *CodeView* manje pouzdano.)

Ne moraju se sve greške pojaviti prilikom prvog ispitivanja programa. Na primer, nekad je potrebno program ponoviti nekoliko stotina puta da bi se dogodila greška. Radi otklanjanja greške, programer menja stanje u memoriji ili tekst programa, ali se time gubi mogućnost reprodukcije. U takvom slučaju dibager bi bio beskoristan da nije mogućnosti da se cela seansa (ili samo njen deo) ponovi. Najvažnija primena dinamičkog ponavljanja programa je upravo mogućnost da programer razmotri stanje pre nego greška nastane.

U opciji **Watch** postavljaju se zaustavne tačke i izrazi čiju vrednost treba proveriti kada se program zaustavi. Postoji pet vrsta zaustavnih tačaka i za svaku se može postaviti po dalj(i)h pet parametara, tako da nema smisla sve to nabrajati, ali se može konstatovati da se *CodeView* sadrži sve što je potrebno.

U opciji **Options** podešavaju se parametri prozora za tekst programa i sadržaj memorije, brzina animacije, način izračunavanja aritmetičkog izraza (jer to zavisi od jezika), razne opcije za dinamičko ponavljanje i sl. Na 386 računarima moguće je razgledati 32-bitne registre i procesorska naredbe.

Opcija **Calls** pokazuje aktivan potprogram i vrednosti lokalnih parametara. Ne pokazuje se ništa sve dok se ne izvrši bar jedan potprogram koji ubacuje podatke na stek.

Postoji osam vrsta prozora i svaki je potpuno nezavistan od ostalih. U jednom se vidi tekst programa, npr. na fortranu ili na asembleru. Dva ovakva prozora istovremeno mogu biti otvorena, za razgledanje istog teksta ili dva različita teksta istovremeno. Prozor sa naredbama prikazuje naredbe koje programer upućuje direktno dibageru (obično su to naredbe kojih nema u menijima). Treća vrsta prozora prikazuje vrednosti određenih promenljivih i izraza. Mogući su direktna izmena vrednosti u memoriji, pregledanje sadržaja složenijih struktura i praćenje pokazivača po memoriji.

Iz naziva prozora za lokalne promenljive, sadržaj memorije, registre, koprocesor i help lako je dokučiti šta prikazuju.

CodeView je odličan.

PROŠIRENJA JEZIKA

Kada se pojavio, fortran kao jezik bio je samo za DO petlju bolji od asemblera. Tek noviji standardi i nestandardna Majkrosoftova proširenja dovode fortran na nivo kojeg odavno imaju paskal i C.

Postoji šest osnovnih tipova u MS FORTRAN-u 5.1. To su INTEGER (*1 ili BYTE, *2 i *4), REAL (*4 i *8), COMPLEX (*8 i *16), LOGICAL (*1, *2 i *4), CHARACTER (nizovi slova od 1 do 32 767 dužine) i slogovi. Postoje i strigovi kao u jeziku C, tj. takvi da se završavaju posebnim znakom za kraj stringa.

Posebno je interesantan pojam složenih promenljivih koji je implementiran po ugledu na C. Postoji nova službena reč STRUCTURE i njome se definiše tip podataka. Takođe novom službenom reči RECORD deklarišu se promenljive na jedan od tipova koji je definisan sa STRUCTURE. Na primer:

```
STRUCTURE /zaposlen/
  CHARACTER*25                prezime
  CHARACTER*15                ime
  INTEGER*2                   broj_kuce
  INTEGER*4                   telefon
  LOGICAL*1                   ozenjen
END STRUCTURE

RECORD /zaposlen/ zaposleni(200)
...
zaposleni(1).broj_kuce=15
...
zaposleni(23).telefon=12345
...
```

Niz zaposleni ima 200 članova, broj kuće prvog je 15, a broj telefona 23-eg je 12345.

Postoje i unije i mape koje – zajedno uzev – proširuju ideju EQUIVALENCE naredbe sa jednostavnih na složene promenljive. Nizovi su takođe prošireni: moguće je deklarisati ih dinamički, tj. tokom izvršenja programa. Dinamički niz deklariše se atributom ALLOCATABLE i brojem dimenzija. Time se samo naznačuje da niz može biti dinamički, a za efektivno stvaranje takvog niza treba izvršiti naredbu ALLOCATE. Postoji i naredba DEALLOCATE koja dinamički niz uklanja iz memorije. Evo primera:

```
INTEGER DATASET[ALLOCATABLE] (:,:,:)
INTEGER REACTOR, LEVEL, ERROR
DATA REACTOR, LEVEL /10, 50/
..
ALLOCATE   (DATASET(REACTOR,LEVEL),STAT=ERROR)
...
DEALLOCATE (DATASET, STAT=ERROR)
...
```

Parametar STAT sadrži informaciju o uspehu zatražene radnje.

Za razliku od drugih jezika, MS FORTRAN dozvoljava operacije nad celim nizovima istovremeno. Na primer, znak + primenjen na dva niza sabira sve članove niza, član po član. Operacije važe i ako je jedan sabirak niz a drugi skalar. Logički i celobrojni nizovi mogu se porediti pomoću operatora .AND. i .OR. itd. Nizovi takođe mogu biti parametri funkcija, a funkcije mogu vraćati nizove kao rezultat. Naravno, da bi sve to važilo, nizovi moraju biti slični, tj. deklarisani na isti način.

Primeri su:

array1 = 42	!Pridodeljuje konstantu svakom elementu
array3 = array1 / array2	!Deljenje odgovarajucih clanova
array2 = – array3	!Negacija svakog clana niza
array1 = array2 – a(7)	!Oduzima konstantu od svakog clana
array3 = MojaF(a2)	!Primenjuje funkciju MojaF na niz a2
ia = .NOT. array2	!Invertuje bitove svakog elementa niza i array2

Ovo pruža divne mogućnosti, a jedina cena je za nijansu sporije izvršenje programa u odnosu na eksplicitno napisane DO petlje.

Treba još spomenuti naredbe AUTOMATIC, CYCLE, EXIT, SELECT CASE, DO WHILE i IF...THEN...ELSEIF...END IF. Naredba AUTOMATIC postavlja promenljivu na stek, a ne zauzima prostor statički (što bi bilo tipično za fortran). Naredba IF je proširena sa ELSE i ELSEIF, a postoji i izbor između niza mogućnosti. Na izbor sintakse C je očigledno imao presudan izbor:

```
CHARACTER*1 CMDCHAR
...
SELECT CASE (CMDCHAR)
CASE ('0')
  WRITE(*,*) „BROJ 0"
CASE('1':'9')
  WRITE(*,*) „IZMEDJU 1 I 9"
CASE('d','D')
  WRITE(*,*) „MALO ILI VELIKO D"
CASE DEFAULT
WRITE(*,*) „NAREDBA NIJE PREPOZNATA'
END SELECT
```

Osim obične DO naredbe, postoji i DO WHILE kao i dve naredbe da se prenebregne pravolinijski redosled izvršenja naredbi iz tela petlje. EXIT napušta DO petlju pre vremena, a CYCLE je isto što i GOTO na sam kraj petlje.

ATRIBUTI

ALLOCATABLE je primer atributa, dodatne informacije koju programer može zadati da bi promenio osobine promenljive tipa, potprograma ili parametra potprograma. Atributi su ALIAS, ALLOCATABLE, C, EXTERN, FAR, HUGE, LOADDS, NEAR, PASCAL, REFERENCE, VALUE i VARYING. Sa ALIAS se zadaje spoljnje ime potprograma ili COMMON bloka, tako da se u jednom delu programa koristi jedno ime a u drugom – neko drugo. Atribut C umesto fortranskih konvencija na potprograme, COMMON blokove i tipove prenosi konvencije iz C-a. Atribut EXTERN označava da se ime nalazi u nekoj drugoj datoteci. FAR označava da se adrese prenose segmentirano. Atribut HUGE kazuje da se formalni argument ili niz može prostrti na više od jednog segmenta. Atribut LOADDS se odnosi samo na posebno prevedene potprograme i funkcije i nareduje prevodiocu da stvori poseban segment za podatke iz tih potprograma i funkcija. Ovaj atribut se uglavnom koristi za OS/2 programe.

Atribut NEAR označava da se podaci nalaze u osnovnom segmentu i da se samo adrese u okviru tog segmenta prenose u potprogram. Atribut PASCAL uvodi u igru paskalske konvencije. Atribut REFERENCE određuje da prenos argumenata bude po imenu, a ne po vrednosti. Nasuprot tome, atribut VALUE znači da se argumenti prenose po vrednosti, što je i normalna praksa za C i paskal. Konačno, broj argumenata u potprogramu ili funkciji ne mora uvek biti isti: atributom VARYING omogućava da broj stvarnih argumenata bude različit od broja formalnih argumenata. Ovo je moguće i u C-u, ali predstavlja mogući izvor grešaka.

METANAREDBE

Program na MS FORTRAN-u osim programskih redova može sadržati i metanaredbe. One počinju znakom $ u prvoj koloni i utiču na način prevođenja programa. Tu se nalaze naredbe za generisanje koda za dibager, uslovno izvršenje delova programa, ubacivanje tekstova drugih programa, broj redova na strani, naslov i podnaslovi programa (za listanje na štampaču) i sl. Posebno je interesantno granuliranje uslovnog prevođenja. Metanaredba $DEBUG prima kao svoj jedini argument string, npr. ovako:

```
C Ovo je komentar jer je slovo C prvo u redu
$DEBUG 'ABCD'
A   I=1
E   I=2
Bl = I + 1
Fl = I + 1
C Ovo je uvek komentar, I=2 jer su izvrsene samo naredbe A i B
```

Ako se jedna od slova iz $DEBUG naredbe nađe u prvoj koloni naredbe, prevodi se kao deo programa. U gornjem primeru, izvršavaju se redovi koji počinju sa A i B, dok je C izuzetak: iako je navedeno u $DEBUG naredbi, ono uvek predstavlja komentar, u skladu sa opštim pravilima fortrana.

Inače, u MS FORTRAN-u 5.1 komentar je i sadržaj reda desno od uskličnika.

FUNKCIJE

Isporučene su čak 132 funkcije, po grupama podeljene ovako: konverzija podataka (realni u celobrojni, slovni u celobrojni, celobrojni u realni itd.), informacije o tipu podataka, zaokruživanje i odsecanje, apsolutna vrednost i prenos aritmetičkog znaka, ostatak deljenja, maksimum i minimum, proizvod brojeva u dvostrukoj tačnosti, kompleksne funkcije, kvadratni koren, eksponenti i logaritmi, trigonometrijske funkcije, tekstualne funkcije, funkcija za kraj datoteke, adresne funkcije, manipulisanje bitovima. Specifična je grupa za informacije o tipu podataka i sastoji se iz sledećih funkcija: ALLOCATED – vraća TRUE ako je memorija pridodeljena nizu, EPSILON – najmanji priraštaj za dati tip podataka, HUGE – najveći broj u okviru zadatog tipa, MAXEXPONENT – najveći mogući eksponent, MINEXPONENT – najmanji eksponent, NEAREST – najbliži sledeći broj, PRECISION – broj značajnih cifara za tip, TINY – najmanji pozitivan broj za dati tip.

Postoje svega tri adresne funkcije: LOCNEAR, LOCFAR i LOC. Prva vraća nesegmentiranu adresu, druga vraća segmentiranu adresu, a treća vraća pokazivač u zavisnosti od odabranog memorijskog modela.

Zastupljene su sve vrste radnji sa bitovima: inkluzivno i ekskluzivno ili, logičko i aritmetičko pomeranje, rotacija, logički proizvod i komplement, testiranje, postavljanje, izmena i brisanje svakog bita ponaosob.

Postoje dodatne grupe potprograma za vreme i datum, greške tokom izvršenja programa, očitavanje parametara sa komandne linije DOS-a, generisanje slučajnih brojeva, poziv funkcija DOS-a, upućivanje signala tj. prekida (interrupt pod DOS-om) i reagovanje na greške matematičkih funkcija.

*Osnovne grupe naredbi u isporučenoj grafičkoj biblioteci su: izbor režima prikazivanja, postavljanje koordinata, paleta boja, vrsta linija i šrafura, crtanje grafičkih elemenata, prikazivanje teksta, snimanje i učitavanje slika. Program koji želi da koristi sve to mora da uveze datoteke FGRAPH.FI i FGRAPH.FD. U prvoj se nalaze INTERFACE TO deklaracije za sve potrebne potprograme i funkcije, a u drugoj su deklaracije struktura, simboličkih konstanti i naredbe EXTERNAL. Naredba INTERFACE TO je slična pojmu prototipa iz C-a: na početku programa se definiše ime, tip i parametarska lista procedure ili funkcije, tako da prevodilac može sam da poredi slaganje „po rodu, broju i padežu" za svako pojavljivanje procedure odnosno funkcije. Tipična deklaracija bi bila:

```
INTERFACE TO INTEGER FUNCTION Fa(p,d,q)
integer*2 p
real d[C]
real*8 q[FAR]
end
```

Biblioteka grafičkih funkcija najviše smisla ima pod DOS-om. Pod OS/2 od grafike nema ništa, ali se bar vide slova, dok pod *Windows*-ima nema nikakvog smisla upotrebljavati isporučenu grafičku biblioteku. Direktno crtanje po ekranu jednostavno bi obrisalo sliku u Windowsima.

FORTRAN POD WINDOWSIMA

MS FORTRAN 5.1 može da proizvede program koji radi pod Windows-ima, ali na jedan ograničen način. Takav program pojavljuje se u posebnom prozoru, ali unutar njega radi se kao na zasebnom DOS-terminalu. Ipak, moguće je razmenjivati podatke preko *Windows Clipboard*-a, pa je zaista tačno ono što se u reklama tvrdi: izračunati podaci iz starog fortranskog programa mogu se prebaciti npr. u *Excel* i tu crtati grafikoni, štampati na svim štampačima koje *Windows* podržavaju itd.

Princip je da jednoj datoteci odgovara prozor pod *Windows*-ima, pa je moguće spremiti podatke u u jednom prozoru, izvršiti fortranski program u drugom, a rezultate naći u nekom trećem prozoru. Dodatna pogodnost je da ako se program prevede za *Windows*-e, onda se memoriji pristupa preko *Windows*-a u standard ili enhanced režimu. To znači da je sva memorija dostupna i to linearno. Nema nikakvih smetnji da se napiše program koji će imati niz od deset megabajta, samo ako toliko slobodnog prostora ostaje za vreme rada *Windows*-a u centralnoj memoriji.

Mogućnost rada fortranskih programa pod *Windows*-ima (a samim tim i pod OS/2 2.0) trebalo bi da odškrine zlatne dveri novih operativnih sistema za sve stare fortranske programe.

JOŠ UVEK IMA SMISLA

Fortran je najstariji „živi" programski jezik. I dalje je prisutan u svesti miliona programera, širom sveta. U ovoj reinkarnaciji koristi DOS okolinu na najbolji mogući način. Očigledno da je C bio uzor za većinu dodataka standardnom jeziku, ali samo po sebi to nije nikakav nedostatak. Zato se sa MS FORTRAN-om 5.1 može PC iskoristiti do kraja: u laboratoriji za prikupljanje podataka „uživo", u poslovnim obradama zahvaljujući stringovima i dinamičkoj alokaciji memorije, a za numeričke potrebe je ionako bez premca. Verovatno će kasnije biti poboljšanja na *Windows* strani, ali sve je to mnogo više od konkurencije koja za sada ne nudi nikakvu *Windows* kompatibilnost.

Ako vam je fortran „maternji" programski jezik, ako imate korisne fortranske programe iz raznih knjiga ili starih listinga, ako hoćete jednostavan jezik kao uvod u programiranje, onda MS FORTRAN 5.1 itekako ima smisla.

ČEKAJUĆI OBJEKTE

Povod za ovaj tekst je jedan mali ali veoma interesantan program. Ipak, opisu biblioteke *o:Clip* ne bismo posvetili previše prostora – reči o svega nekoliko novih stvari – da nas nije podstakao da, na najelementarnijem nivou, objasnimo i osnovne pojmove objektnog programiranja. Dati su najjednostavniji mogući primeri – ako vam zatrebe, sami se potrudite da ih zakomplikujete!

Nenad Batoćanin

U *Clipper 5.0* je, kao što smo već pisali, ugrađena ograničena mogućnost korišćenja objekata upotrebom unapred definisanih objekata možete vršiti tabelarni prikaz datoteka i nizova, učitavati vrednosti promenljivih i kontrolisati greške u programu. Dodavanjem ovih pogodnosti – što je samo po sebi veliki napredak – *Clipper* ipak nije postao pravi objektni jezik. Da bi jedan programski jezik podržavao objektno programiranje (OOP), mora da zadovoljava određene uslove. Doduše, većina savremenih OOP jezika ne zadovoljava baš sve teorijske uslove, ali minimum mora da postoji: korisniku se mora omogućiti bar kreiranje novih objekata. *Clipper 5.01* to ne zadovoljava: postoji samo nekoliko ugrađenih tipova objekata koje možete koristiti. Ovakvo rešenje prilično je neobično: ako je urađen sistem za podršku nekoliko tipova (ugrađenih) objekata, zašto ne postoji i mogućnost njihovog kreiranja? Bilo kako bilo, ni verzija 5.01 nije donela ovu mogućnost. Na prvi pogled, ni verzija 5.01 nije donela ništa značajno što se tiče objekata. Međutim, prava istina je da je *Nantucket* u toj verziji ugradio poziva za prošireni rad sa objektima, ali to, izgleda, nije završeno do kraja, pa nije ni dokumentovano. Ovo je vrlo važna informacija – to znači da će sledeća verzija *Clipper*-a skoro sigurno imati sve mogućnosti OOP-a.

Nedokumentovani sistemski pozivi za rad sa objektima i sada se mogu iskoristiti: postoji više različitih programa koji omogućavaju da korisnik kreira svoje sopstvene klase. Minijaturna biblioteka *o:Clip* je jedan od tih programa: ona vam omogućava da kreirate svoje objekte i da ih, zatim, koristite potpuno ravnopravno sa objektima klase **TBrowse**, **Get** ili **Error**. I sve to uz minimalnu cenu: *o:Clip* zauzima svega oko 3K memorije pri izvršavanju.

KO SE BOJI OBJEKATA

Bez obzira na sve prednosti OOP-a, mnogi programeri još uvek izbegavaju korišćenje objekata u svojim programima. Obično se radi o nepoverenju u nove stvari, o urođenoj inertnosti i slično. Treba reći i da programeri kod nas nisu baš previše informisani o toj temi. Neupućeni obično misle da se radi o nekom složenom i egzotičnom sistemu koji bez ikakve potrebe komplikuje stvari i, u suštini, „nije ni za što". S obzirom da su naši programeri uglavnom odrasli uz računare skromnih mogućnosti, težnja da ostanu što bliži mašini im je zapisana u genetski kod. Zato svaka komplikacija koja ih udaljava od mašine biva dočekivana „na nož".

Ubediti ljude da OOP nije komplikacija, već pojednostavljenje programa, zaista nije lako. Jer, ako je neko video neku „fabričku" hijerarhiju objekata (*Turbo Vision*, na primer), verovatno je u prvi mah bio zastrašen velikim stablom i kako sad to može biti jednostavno? Ipak, OOP je **pojednostavljenje** programa, ma kako vam to čudno izgledalo. Radi se o reorganizaciji koda po jednostavnim principima, čime se postiže da se programi brže i lakše pišu, jednostavnije održavaju, itd.

Ako malo analizirate bližu istoriju softverske tehnologije, primetićete da je, u suštini, mnogo više pažnje bilo posvećivano krajnjem korisniku programa, nego programeru. To je u neku ruku i logično, jer programi su i namenjeni korisniku, a ne programeru. Tako su razvijeni složeni korisnički interfejsi, potpuno podređeni korisniku. Pisanje programa u takvom okruženju za programera može biti veoma zamorno. Sa druge strane, zahtevi korisnika su sve više rasli, pa savremeni programi mogu biti prava čudovišta po složenosti. Održavanje takvih programa je izuzetno ozbiljan i težak posao. OOP je, posle dužeg vremena, prvi korak da se olakša posao programeru, a ne samo korisniku (mada se i ovo drugo ne isključuje). Većina trenutno aktuelnih programskih jezika pravi veoma vidljiv zaokret prema OOP-u.

o:Clip ZA POČETNIKE

Osnovni principi u OOP-u su veoma prosti. Bez teorijskog uvoda, **objekte** za početak možete sasvim ispravno shvatiti kao **slogove** sasvim slične tipu **RECORD** u *Pascal*-u. Jedina razlika je što ovi slogovi mogu sadržati kao elemente i funkcije, što je veoma značajno, kako ćemo kasnije videti. Za početak možemo definisati jedan jednostavan objekat-slog koji predstavlja podatke za neku osobu:

```
CLASS Person
    VAR Name, Address, Phone
ENDCLASS
```

Kada se u programu pojavi ovakva deklaracija, kaže se da smo definisali jednu **klasu objekata** (koja se za naš primer zove **Person**). U terminologiji „standardnih" programskih jezika, mi smo ovom deklaracijom, u stvari, definisali novi tip podataka: slog sa poljima **Name**, **Address** i **Phone**. U OOP-u se ovo nešto drugačije zove: tip podataka se naziva **klasa objekata**, a svaki element te klase (u stvari, konkretan slog) **objekat**. Dobro to upamtite – ovakva terminologija je uobičajena u svim tekstovima o OOP-u. Ovi nazivi su izabrani po analogiji i trebalo bi da doprinesu lakšem razumevanju stvari, ali čini mi se da je kod nas upravo suprotno.

Kako se sad koriste takvi objekti? Pošto *Clipper* nema tipizaciju promenljivih, ne može se deklarisati neka promenljiva tipa **Person**. Umesto toga, dovoljno je pozvati funkciju koja će kreirati novi slog (objekat) i dodeliti ga nekoj promenljivoj. Funkcija za kreiranje objekta ima isti naziv kao i klasa objekata. Na primer,

```
oStudent := Person()
```

kreira novi objekat-slog i dodeljuje ga promenljivoj sa nazivom **oStudent**. Analogno se može kreirati proizvoljan broj objekata sa različitim nazivima. Kreirani objekti se mogu dodeljivati različitim promenljivim ili elementima nizova:

```
oProfessor := Person()
LOCAL oArray[100]
aFill (oArray, Person())
```

Svaki poziv funkcije **Person** odvaja potreban memorijski prostor i kreira jedan novi objekat iz klase **Person**, koji ima tri svoja polja: **Name**, **Address** i **Phone**. Naravno, **svaki** objekat ima svoja polja **Name**, **Address** i **Phone**, koja nemaju nikakve veze sa ostalim.

Sledeći korak je korišćenje tih slogova. Polja sloga su, u stvari, sasvim obične promenljive koje se mogu koristiti baš kao i sve druge. Jedina razlika je što **pripadaju** jednom objektu, i da bi im se pristupilo, mora se navesti njihov „vlasnik", tj. objekat koji ih „sadrži". Za pristupanje elementima objekta koristi se operator „:" (dvotačka). Na primer, **oStudent:Name** označava polje (promenljivu) **Name** objekta **oStudent**. Kao što je rečeno, ovakve promenljive su potpuno ravnopravne sa drugim promenljivim. Može se pisati, primera radi:

```
oStudent:Name := "Pera Peric"
```

čime se promenljivoj **Name** objekta **oStudent** dodeljuje string "Pera Peric". Naravno, ove promenljive se mogu upotrebljavati i u izrazima, pa se može pisati i:

```
QOut (PadR (oStudent:Name, 50))
cRet := oStudent:Address + " Beograd"
```

itd.

Na žalost, da bi sve to radilo, *o:Clip* zahteva neka ograničenja. Prvo, na početku datoteka gde se definišu klase mora se uključiti header datoteka **OCLIP.CH**, koja sadrži pretprocesorske direktive koje omogućavaju korišćenje novih naredbi za definisanje novih klasa. Znači, direktiva **#include "oclip.ch"** mora uvek biti na početku svakog programa. Drugo, zbog načina na koji se interpretira definicija klase (pretprocesira se u funkciju), definicija klase ne sme biti na početku prve datoteke koja se linkuje u EXE program, jer će biti shvaćena kao glavni program. Najbolje je klase definisati u posebnoj datoteci ili posle svih procedura. Najgore je što može biti sintaksnih problema oko uklapanja objekata u nekim naredbama (**GET**, na primer).

PORUKE I METODI

Kao što je rečeno, objekti se mogu shvatiti i kao generalizovani slogovi: osim „običnih" polja koja čine slog, objekat može sadržati i funkcije (odnosno procedure) kao elemente. To je i glavna razlika: umesto grupe podataka, sad možemo napraviti grupu podataka i **funkcija**. Za prethodni primer može se napraviti procedura **Print**, koja formatira i štampa elemente datog objekta. Takva procedura se može jednostavno napisati i bez korišćenja OOP-a: napiše se procedura **Print(s)**, gde je **S** objekat čiji se elementi štampaju.

Ipak, u OOP-u se obično koristi drugo rešenje. Procedura **Print** se dodaje kao novi element objekta, kao tzv. **metod**. Na slici 1. je prikazan primer definisanja klase objekata **Person**. Svaki objekat sadrži i funkciju za štampanje svog sadržaja. Naziv funkcije se deklariše posle reči **METHOD**, a funkcija se stvarno definiše ispod toga. Ovako definisana funkcija je deo svakog objekta tipa **Person** i ne može se koristiti izvan njega, za razliku od standardne funkcije **Print**. Na žalost, *o:Clip* ovde pokazuje priličnu slabost: funkciju **Print** možete upotre-

```
#include 'oclip ch'

PROC Main
   // Kreiranja novog objekta
   s := Person()

   // Dodeljivanja vrednosti promenljivim
   s:Name    = 'Pera Peric'
   s:Address  = 'ul. Kestenova 46'
   s:Phone   = '223 664'

   // Stampanje podataka
   s:Print()

RETURN

CLASS Person
   VAR Name, Address, Phone
   METHOD Print
ENDCLASS

PROC Print
   ? 'IME         ', ::Name
   ? 'ADRESA   ', ::Address
   ? 'TELEFON  ', ::Phone
RETURN
```

Primer definisanja metoda

```
// Bafer za stek i pokazivac
STATIC Buffer[100], StackPtr

PROC Main
   StackInit()
   Push ('A')
   Push ('B')
   Push ('C')
   ? Pop ()
   ? Pop ()
   ? Pop ()
   ? Pop ()
RETURN

// Inicijalizacija steka
PROC StackInit
   StackPtr := 0
RETURN

// Upis elementa
PROC Push (x)
   IF StackPtr < 100
      Buffer[++StackPtr] := x
   END IF
RETURN

// Citanje elementa
FUNC Pop
   LOCAL Ret

   IF StackPtr<>0
      Ret := Buffer[StackPtr--]
   ELSE
      Ret := NIL
   END IF
RETURN Ret
```

Klasične realizacija steka

```
#include "oclip ch"

PROC Main
   S := Stack ()
   S:init ()

   S:Push ("A")
   S:Push ("B")
   S:Push ("C")

   ? S:Pop ()
   ? S:Pop ()
   ? S:Pop ()
   ? S:Pop ()

RETURN

// Definicija klase objekata
CLASS Stack
   VAR Buffer, StackPtr
   METHOD init, Push, Pop
ENDCLASS

// inicijalizacija steka
PROC Init
   ::StackPtr := 0
   ::Buffer := {}
   aSize (::Buffer, 100)
RETURN

// Upis elementa
PROC Push (x)
   ::Buffer[++::StackPtr] := x
RETURN

// Citanje elementa
FUNC Pop
   LOCAL Ret

   IF ::StackPtr<>0
      Ret := ::Buffer[::StackPtr--]
   ELSE
      Ret := NIL
   END IF
RETURN Ret
```

Stek realizovan preko objekata

biti i samostalno (što se ne bi smelo dozvoliti), ali će to sigurno prouzrokovati grešku. Ovo je razumljivo, jer mala zbirka funkcija kao što je *o:Clip* nema previše uticaja na osnovne mehanizme *Clipper*-a.

Kako u funkciji **Print** pristupiti pojedinim promenljivim? Pošto je **Print** funkcija kao i svaka druga, za sva „normalne" promenljive važe uobičajena pravila. Problem su samo promenljiva koje pripadaju istom objektu kome pripada i **Print**. Tim promenljivim se pristupa korišćenjem operatora „::". Taj operator označava promenljivu koja pripada upravo tom objektu. Na primer **::Name** označava promenljivu **Name** tog objekta.

Iz prethodnog primera se ne vide nikakve prednosti OOP-a nad standardnim pristupom. Čak može vam izgledati da je sve to nepotrebno komplikovanje. Međutim, u narednim primerima možete videti da to nije tako.

Ovakav rad sa objektima je u suštini veoma prirodan. Imete mogućnost da definišete neka svoje „robote" koji znaju da „rade" određene operacije i zatim „upravljate" njima. U prethodnom primeru **Person** je opis jednog „tipa robota", a S jedan primerak koji smo kreirali. Taj „robot" zna samo da štampa svoj sadržaj. To mu nalažemo „komandom" **Print**. Zato se kaže da je naredba **s:Print()** **slanje poruke objektu**. Pri tome je **S** naš robot-objekat, a **Print** poruka koju mu šaljemo. Zato se implementacija procedure **Print** zove **metod** – to je način na koji „robot" izvršava naređenja. Ova analogija ima svoje puno opravdanje: kao što je kreiran objekat **S**, može se kreirati i proizvoljan broj drugih objekata, i svakim se može upravljati tako što mu se šalju određene poruka.

Ovde se pojavljuje jedan praktičan problem koji nema direktne veze sa OOP. Naime, šta ako dve klase objekata imaju funkciju (metod) sa istim nazivom? Primera radi, i neka klasa **Window** može imati metod **Print**. U pravom OOP-u tu nema nikakve dileme: ove dve procedure nemaju baš nikakve međusobne veze, jer svaka pripada svojoj klasi objekata. Pošto je *o:Clip* ipak „skrpljeno" rešenje, ovako nešto nije moguće. Zato je omogućeno da se pri definiciji metoda posebno zadaju naziv metoda, a posebno naziv funkcije koja ga realizuje. Na primer, može se pisati **METOD Print – PersonPrint**. Sada objekat takođe sadrži metod **Print**, ali se on zadaje procedurom po nazivu **PersonPrint** – na taj način je omogućeno da više različitih klase imaju metode istog naziva. Naime, sada se poruka **Print** može slati i objektima iz klase **Person** i klase **Window**, ali će se aktivirati različite procedure, u zavisnosti kom objektu je poslata poruka.

OBJEKTNO PROGRAMIRANJE

Vratićemo se za trenutak na osnovnu ideju OOP-a. Razni programski jezici koriste različite **modele** programiranja. Na osnovu toga se svi programski jezici dele na grupe. Na primer imperativni jezici, funkcionalni, logički, objektno orijentisani. Svaka od tih grupa ima specifičnu logiku i način izvršavanja zadatih naredbi (ili nekog ekvivalenta naredbi). Naravno, najpopularnija i najrasprostranjenija od svih grupa su imperativni jezici. Kod njih se program izvršava na svima poznat način: po nekom redosledu se izvršavaju naredbe koje menjaju određene promenljiva i tako na osnovu ulaznih pod rešenja nema neku očiprograma u funkcionalnom programiranju svodi se na izračunavanje vrednosti raznih funkcija: kod logičkog programiranja se program zadaje u vidu logičkih relacija na osnovu kojih se dobijaju rezultati. OOP se takođe može posmatrati kao jedan takav model: izvršavanje programa se svodi na određenu komunikaciju između objekata. Objekti šalju poruka drugim objektima: pri tome se i klase objekata i poruka mogu shvatiti kao svojevrsni objekti. Znači, potpuna implementacija OOP-a ne traži nikakve „pozajmice" od ostalih vrsta programiranja. Ovakvi koncepti su primenjeni u relativno „čistim" objektnim jezicima (*Smal-*

Listing 4

```
#include "oclip ch"

PROC Main
   // Korišćenje klase Stack celoje lele
   S  = Stack ()
   S Init ()

   S Push ("A")
   S Push ("B")
   S Push ("C")

   ? S Pop ()
   ? S Pop ()
   ? S Pop ()
   ? S Pop ()

   // Kreira se objekat nove klase
   S  = ScreenStack()

   S Init()

   S Push("A")
   S Push("B")
   S Push("C")
   S Push("D")
   S Display()

RETURN

// Definicija klase objekata
CLASS Stack
   VAR Buffer, StackPtr
   METHOD Init, Push, Pop
ENDCLASS

// Inicijalizacija steka
PROC Init
     StackPtr  = 0
     Buffer := {}
   aSize ( Buffer, 100)
RETURN

// Upis elemenata
PROC Push (x)
     Buffer(++ StackPtr)  = x
RETURN

// Citanje elemente
FUNC Pop
   IF  StackPtr<>0
     Ret := Buffer[ StackPtr-- ]
   ELSE
     Ret  = NIL
   END IF
RETURN Ret


// Nova klasa, 'dete' klase STACK
CLASS ScreenStack FROM Stack
   VAR Visible, Line
   METHOD Display, Init = ScreenInit
ENDCLASS

// Ispisivanje steka
PROC Display
   LOCAL i

   IF  Visible
     @  Line, 0
     FOR i := 1 TO  :StackPtr
       @  Line, i SAY  Buffer[i]
     NEXT
   END IF
RETURN

// Inicijalizacija objekta klase SCREENSTACK
PROC ScreenInit
   ::Parent:Init()

   ::Visible := .T.
   ::Line  = 24
RETURN
```

Nasleđivanje klase objekata

Listing 5

```
PROC Box (x1, y1, x2, y2, Color)
   LOCAL oldColor  = SetColor (Color)
   @ x1,y1 CLEAR TO x2,y2
   @ x1,y1 TO x2,y2 DOUBLE
   SetColor (oldColor)
RETURN
```

Klasična procedura za crtanje prozora

Talk, na primer) Svaki od navedenih koncepata ima svoje prednosti i mane. Sasvim je sigurno jedno – danas apsolutno preovlađuju imperativni jezici

Međutim, kako je za uspeh jednog jezika važna ne samo teorijska zasnovanost, nego i praktična primena, čak i tipični predstavnici objektnih jezika kao što je *SmallTalk* imaju „imperativne" elemente. Moguće je i obrnuto prebaciti OOP u imperativni jezik. Time se neke prednosti OOP-a dobijaju u imperativnom jeziku i tako vam je omogućeno da OOP koristite u onoj meri koliko vam odgovara. Tako su nastali razni hibridni jezici, od kojih je najpoznatiji *C++*

Osnovnu prednost OOP-a nije tako lako objasniti na primeru. Prava prednost se vidi u veoma složenim programima koji su naporni za analizu, a kod jednostavnih primera se često gubi osnovni smisao. Naredni primer je prilično jednostavan, ali zadržava osnovnu ideju.

Sigurno vam je u programu često zatrebala neka struktura podataka u obliku steka. Na primer, pravite sistem za korisnički interfejs, pa na steku pamtite „pozadine". Teško je pogoditi neki univerzalni način razmišljanja, ali prilično je izvesno da se bez korišćenja objekata taj problem uobičajeno rešava kao na slici 2. Globalni niz **Buffer** služi za čuvanje vrednosti, a promenljiva **StackPtr** je indeks poslednjeg upisanog elementa. Procedure **Init** i **Push** služe za inicijalizaciju i upis vrednosti na stek, a funkcija **Pop** „skida" poslednju upisanu vrednost sa steka. Ekvivalentno rešenje koje koristi objekte prikazano je na slici 3. Tu se stek definiše kao klasa objekata, od kojih svaki ima svoje promenljive **Buffer** i **StackPtr**. Možete se sami uveriti da su procedure i funkcije (metodi) relativno slične kao i u prethodnom primeru. Jedino je procedura za inicijalizaciju za nijansu složenija, ali to je zbog tehničkog ograničenja *o Clip* biblioteke, a ne zbog OOP-a.

U suštini, ni jedno od ova dva rešenja nema neku očiglednu suštinsku prednost. Međutim, stvar postaje bitno različita kada se rad nastavi. Primera radi, zatreba vam da napravite još jedan stek – za čuvanje pozicija kursora, na primer. Klasičan primer je tu veoma nepogodan za izmene. Da bi se uveo stek, mora se prvo uvesti još jedan par promenljivih **Buffer** i **StackPtr**. Još gore, moraju se menjati sve procedure koje rade sa stekom, kako bi se označilo na koji stek se tekuća operacija odnosi. Tu postoje dva načina: pisanja novih procedura (na pr. **Init2**, **Push2**, **Pop2**) koje bi operisale samo sa drugim stekom, ili izmena procedura tako da **Buffer** ne bude globalna promenljiva, već da se prenosi kao argument. Tada bi se podaci upisivali sa

Push (Buffer2, "A")

I jedno i drugo rešenje je veoma loše, iz mnogo razloga. U svakom slučaju, program mora da trpi značajne i neprijatne izmene.

Objektno rešenje sad pokazuje svoju veliku prednost. Ako vam zatreba još jedan stek, dovoljno je da napišete

P := Stack()

i time dobijate novi stek **P**. Potpuno iste procedure sad se koriste za rad sa novim stekom. Na primer, može se pisati

S:Push ("A")
P:Push ("B")

Dalje uopštavanje je kod standardnog postupka još gore: zamislite da vam treba niz stekova čiji broj određuje korisnik za vreme izvršavanja programa! Čak i da nekako prepravite program, rezultat će verovatno biti toliko „iskrpan", da eventualna dalja proširenja neće imati nikakve šanse (matrica stekova, na primer). Takva građevina nema šanse ni prilikom izmene temelja: ako vam se učini da niz **Buffer** ne odgovara za implementaciju steka i zamenite ga nečim drugim, skoro sigurno će se cela građevina srušiti ili bar opasno naknviti.

Sa objektnim rešenjem su takve nadgradnje izuzetno jednostavne i, što je najvažnije, zahtevaju minimalnu izmenu programa. Na primer, ovako se definiše niz stekova

LOCAL aStacks[100]
aFill (aStacks, Stack())

Proizvoljna nadgradnja nad osnovnim stekom nikako se ne odražava na definiciju steka. Sa druge strane, kasnija eventualna izmena te definicije se ne mora odražavati na program koji je koristi. Na taj način se uvode slojevi koji su međusobno nezavisni, što je za strukturu programa izuzetno dobro.

NASLEĐIVANJE

Jedna od osnovnih koncepcija u OOP je tzv. **nasleđivanje** klasa. To je jednostavan i elegantan način da stvaramo nove objekte bazirane na već postojećim. Na primer, pretpostavimo da nam je potreban stek u programu, ali da se pri tom prikazuje njegov sadržaj na ekranu. Ovo se može uraditi na više načina, ali samo neki su u duhu OOP-a. Najlošije rešenje je da napravimo proceduru koja koristeći interne promenljive objekta **S** ispisuje elemente steka. Nešto kvalitetnije rešenje je da dodamo definiciji klase **Stack** metod **Display**, koji služi za ispisivanje vrednosti u steku. Uradićemo to na treći način: nasleđivanjem.

Naime, nama treba objekat potpuno isti kao **Stack**, ali još da zna da se „nacrta". To može-

Listing 6

```
#include "oclip.ch"

PROC Main
   // Kreiranje dva prozora
   A := Box() , B := Box()

   // Podesavanje parametara
   A:SetXY(0,0,7,20) , A:SetCol("n/w")
   B:SetXY(8,50,18,70) , B:SetCol("w/n")

   // Ispisivanje na ekranu
   A:Display()
   B:Display()

   // Paralelno se prikazuju podaci sa dva perif. uredjaja
   WHILE IsDeviceOK()
      A:Type(GetData(1))
      B:Type(GetData(2))
   END

   // Zatvaranje prozora
   A:Close()
   B:Close()
RETURN

// Definicija klase objekata
CLASS Box
   VAR Screen, x1, y1, x2, y2, Color
   METHOD Display, Type, Close, SetCol, SetXY
ENDCLASS

// Podesavanje koordinata
PROC SetXY (ux1, uy1, ux2, uy2)
     ::x1 := ux1
     ::y1 := uy1
     ::x2 := ux2
     ::y2 := uy2
RETURN

// Ispisivanje prozora
PROC Display
   LOCAL oldColor := SetColor(::Color)

   ::Screen := SaveScreen (::x1, ::y1, ::x2, ::y2)
   @ ::x1, ::y1 CLEAR TO ::x2, ::y2
   @ ::x1, ::y1 TO ::x2, ::y2 DOUBLE
   SetColor (oldColor)
RETURN

// Zatvaranje prozora
PROC Close
   RestScreen (::x1, ::y1, ::x2, ::y2, ::Screen)
RETURN
// Ispisivanje stringa u prozor
PROC Type (Text)
   LOCAL x,y
   x := (::x2 - ::x1) / 2 + ::x1
   y := (::y2 - ::y1 - Len(Text)) / 2 + ::y1
   @ x,y SAY Text COLOR ::Color
RETURN

// Podesavanje boje
PROC SetCol (UserColor)
   ::Color := UserColor
RETURN
```

Procedure za rad sa prozorima korišćenjem objekata

mo uraditi tako što ćemo izmeniti definiciju klase **Stack**, ali zašto je dirati? Ako nju izmenimo, rizikujemo izmenu dela programa koji već koristi klasu **Stack**, što nikako nije dobro. Zato ćemo definisati „dete" klase **Stack** koje će zadržati sva osobina ove klase, ali će znati i da se „ispiše" na ekranu. Rešenje je dato na slici 4. Nova klasa **ScreenStack** je definisana kao naslednik klase **Stack**, što znači da se sva interne promenljive i metodi klase **Stack** direktno nasleđuju u klasi **ScreenStack**. To je zadato konstrukcijom

CLASS ScreenStack FROM Stack

Na ovaj način smo dobili traženu klasu, iskoristili smo pri tom već postojeću, a da je pri tome nismo ni na koji način izmenili. Primetite da smo za klasu **ScreenStack** definisali nov metod **Init** (implementiran kao procedura **ScreeninIt**) koji osim stare obavlja i neka operacije „nove" inicijalizacije. Pri tom nije potrebno ponavljati naredbe napisane u proceduri **Init** za inicijalizaciju staka – dovoljno je upotrebiti konstrukciju ::**Parent** koja označava „objekat – roditelj" i pozvati njegovu funkciju za inicijalizaciju. Primetite jedan vrlo važan detalj: pri definisanju klase **ScreenStack** vi niste ni na koji način morali da znate kako izgleda definicija klase **Stack**. Pri tome se izuzimaju interna promenljive **Buffer** i **StackPtr**, a i one su uzete zbog pojednostavljenja primera. Dovoljno je bilo znati samo **kako** taj objekat „radi". Principijelno je bilo moguće praviti ovakvu nadgradnju i da smo posedovali samo OBJ datoteku klase **Stack**. Tako se i isporučuju biblioteke klase: lep primer je **TBrowse** koji je ugrađen u *Clipper* – možete ga odlično kontrolisati i praviti svoju nadgradnju, bez listinga na koma je on definisan.

Nasleđivanje se, naravno, može i nastaviti: tako se gradi jedna hijerarhija klasa koju možemo koristiti po potrebi i u drugim programima.

OBJEKTNI INTERFEJSI

Jedna od oblasti gde se objekti često koriste je kreiranje korisničkih interfejsa. Ukratko, objektno orijentisan korisnički interfejs je mnogo fleksibilniji, jednostavniji za programiranje i moćniji u krajnjem rezultatu.

Na primer, kreiramo korisnički interfejs koji će sa korisnikom komunicirati preko sistema prozora. Naravno, pri tome se javljaju razni (potencijalno veoma ozbiljni) problemi. Osnova ovog sistema će, svakako, biti neka procedura za otvaranje i zatvaranje prozora. U zavisnosti od naše osnovne ideje, može se reći da se to svodi na ispisivanje nekog pravougaonika na ekranu. Osnova takva procedura može biti kao na slici 5. Kao argumenti se zadaju koordinate uglova i boja pozadine. Na žalost, ovo je trivijalni deo problema. Mnogo veći problem nastaje kad treba, na primer, ispisati istovremeno više prozora na ekranu. Prvo, potrebno je da svaki prozor memoriše pozadinu koju „uništava" svojim ispisivanjem, kako bi se kasnije restaurirao ekran. Obično se za to koristi stek u koji se upisuju odgovarajuće vrednosti. Sledeći problem je kako organizovati ispis teksta u prozor. Najjednostavnije (i najlošije) rešenje je da se u programu napiše nešto poput

X := 2 , Y := 5
Box (X,Y, X+3,Y+40, "w/b")
@ X+1,Y+2 SAY "Uključite stampac!"

Znači, na osnovu koordinata na kojima je ispisan okvir, pišemo odgovarajuće koordinate za ispis. Ovo rešenje je krajnje loše: uvek kada se u programu pojavljuju neposredne koordinate, stalne prepravke su vam suđene. Moguće je napraviti neku univerzalnu proceduru **BoxType** za ispis teksta u aktivan okvir, pri čemu će procedura **Box** na neki način „javljati" gde je ispisana proceduri **BoxType**. Međutim, sve je to ipak beskonačno komplikovanje.

U prethodnom primeru se javlja još nešto: boja kojom će se ispisati tekst poruke. Procedura **Box** menja boju samo privremeno, a zatim je vraća na staru, tako da o boji ispisa moramo voditi računa posebno. Vidi se da je idealno da procedura **Box** sama ispisuje tekst. Ona ima sve potrebne informacije: i pozicije na ekranu i boju itd. Ovakvo razmišljanje nameravno vodi kreiranju procedure koje bi se pozivala približno ovako

Box (x1,y1,x2,y2,Color,Text, ...)

Spisak nezadrživo raste kako nadgrađujete sistem, i što je najgore, svaka izmena se odražava i na tekst programa koji ste ranije napisali. Čak i razrađen, ovakav sistem veoma teško izlazi na kraj sa nekim problemima, kao što je ovaj: na primer, tekst treba ispisivati, ali istovremeno u više otvorenih prozora (na primer, dobijaju se naizmenično podaci sa više periferijskih uređaja). Tu cela koncepcija funkcije sa velikim brojem argumenata pada u vodu, jer se priroda rešenja ne slaže sa ovom koncepcijom. Lep (u stvari, ružan) primer za ovo je funkcija **dbEdit** ugrađena u *Clipper*: njeno korišćenje je veoma komplikovano i podložno greškama. Zamislite tek slučaj da se izmene argumenti te funkcije!

Objekti ovde pružaju izvanredno elegantno rešenje. Jedan primer prikazan je na slici 6. Prozor se definiše kao jedan objekat. Definisane su metode za razna podešavanja tog objekta (koordinata i boja). Pošto se kreira novi objekat, prvo se podese razni parametri. Zatim se šalje poruka **Display** koja mu nalaže da se „nacrta" na ekranu. Slično se aktiviraju i ostali objekti-prozori. Zatim se porukama **Type** naređuje ispisivanje teksta u izabranom prozoru. Na kraju poruka **Close** zatvara prozor. Kao što vidite, upravljanje objektima je veoma lagano. Kreiranje proizvoljnog broja prozora tako postaje skoro trivijalan posao.

NE MOŽE SE IZBEĆI

Objektni jezici su veliki napredak u programiranju i predstoji im svetla budućnost. Standardni *Clipper* programi su takvi da zahtevaju česte izmene, pa bi uvođenje OOP-a u ovaj posao bilo zaista prava stvar. To bi se najbolje osetilo kod velikih programa.

Sledeća verzija *Clipper*-a će skoro sigurno podržavati OOP na višem nivou (biće moguće kreirati sopstvene klase). U tom smislu je *o:Clip* kao i slična ostvarenje korak u pravom smeru. Upotreba ovih biblioteka bi svakako doprinela lakšem pisanju i održavanju programa. Međutim, tu se javlja jedan veliki problem: kompatibilnost sledeće verzije *Clipper*-a. Situacija je slična kao i sa bibliotekama za *Clipper*. Svaka nova verzija *Clipper*-a zahtevala je i nove biblioteke. Najgore je što nam niko ne garantuje da će *Nantucket* uvesti istu sintaksu kao *Peter Freese*, autor *o:Clip*-a. Čak, skoro je sigurno da neće. I onda sav naš trud pada u vodu: svi stari programi se moraju ručno prepravljati kako bi proradili pod novom verzijom.

Međutim, bez obzira na to, *o:Clip* svakako treba bar probati: njegovim korišćenjem možete uz minimalnu cenu isprobati suštinski nova i moćne koncepte. Utoliko pre što ga veoma lako možete preuzeti sa *Sezama*.

BRAK IZ RAČUNA

Ovih dana se navršave sedam godina od lansiranja prvog pravog 32-bitnog mikroprocesora – Motorola 68020. Tokom tih sedam godina emanila au sa tri generacije 32-bitnika, a atvoreni eu i prvi 32/64-bitni i 64-bitni procesori. Vreme 32-bitnika takode je obeleženo i obračunom CISC i RISC filozofije, koji ae završava pomirenjam sa nakom vretom apajanja dve auprotnosti u novi, viši kvalitat.

Velika zasluga za veoma dinamičan razvoj 32-bitnog sveta pripada žastokoj konkurenciji mnogobrojnih manja i više velikih mikroalektroničkih firmi koje su se upustile u bitku za ovo zahvalno tržište. Pored svima znanog Intela sa X86, 860 i 960 porodicama, u igri su i Motorola (680×0, 881×0), AMD, Chips&Tech i partneri u anti-Intel konzorcijumu (X86 kompatibilci), IBM (POWER RISC), NS (32xxx), AMD (290×0), Sun i partneri (SPARC), SGI/MIPS i partnari (Rx000), Inmos (T8xx, T9000), HP (Precision, Snake) i, najsvežiji, DEC sa izuzetno zanimljivom „Alpha" arhitekturom. Naravno, prisutno je i nekoliko drugih firmi, ali pomenuta su najzanimljivija za nas.

POJAČANJE IZ KUĆE

"Intel", firma u kojoj je nastao prvi mikroprocesor i nikli ključni ljudi mnogih drugih IC firmi, godinama je slavu (i monopol na PC tržištu) gradio na jednoj porodici procesora, 80×86 porodica, kojoj je rodonačelnik stari 8086, svoj uspeh je (nazasluženo) zasnovala na uspehu IBM PC standarda. Da je svojevremeno IBM izabrao 68000 porodicu za prvi PC, nikada ne bismo imali muka sa 640 K granicom, segmentiranjem, DOS-om, OS/2 itd. Srećom, deo basnoslovnog bogatstva zgrnutog na procesorima za PC „Intel" je usmeno na razvoj dve nove porodice mikroprocesora čiji su prvi predstavnici izašli na svet 1989. Prva, i960, je porodica RISC procesora specijalne namene (pretežno kontroleri posebne nemene), a druga je i860 porodica vrlo brzih RISC procesora, optimizovanih za brzu matematiku i 3-D grafiku.

"Intelov" prvi 860, lansiran nekoliko meseci pre 486, doživeo je prilično dobar uspeh. Za to je najviše zaslužna mogućnost ultrabrzog vektorskog računanja u pokretnom zarezu, kao i ugrađeni 3D grafički procesor. Ova prednosti su, bez obzira na inače dobre karakteristike ostalih delova procesora koji daju osnovu za veoma snažnu UNIX mašinu, u startu usmerile i860 porodicu u svet kompleksne matematike i brze grafike na PC i svim klasama radnih stanica. Od high-end kartica za PC do najbržih HP, SUN i NEXT radnih stanica, srce grafike svuda je 860. Od ostalih predstavnika svetske RISC porodice ovaj procesor odvaja i laka povezivost sa x86 arhitakturom, sa kojom može da radi u simbiozi i to upravo na onim poslovima u kojima se Intelovi procesori najlabiji.

Veoma bitna odlika 860 porodica su identični tipovi podataka i stranični mehanizam za upravljanje memorijom kao i kod 386/486 porodice. MMU (Memory Management Unit) deo je procesora zadužen za stranično upravljenje memorijom, u koje spadaju translacija logičkih (programskih) u fizičke (stvarne, na spoljnoj adresnoj sabirnici) adrese, zaštita pojedinih stranica od upisa itd. To, pored sličnih tajminga spoljnih sabirnica, omogućava koegzistenciju 486 i 860 na zajedničkoj sabirnici uz, istini za volju, malo složeniju logiku koja potire fizičke razlike spoljnih sabirnica dva procesora (Hauppauge 4860 ploča).

Nebojša Jovanović

"Intelova" generalna strategija za ovu deceniju je spajanje x86 i 860 procesora u jednu fizičku celinu, s postepenim prebacivanjem x86 softvera na snažniji RISC i, u skladu s tim, postepeno nestajanje x86-kompatibilne jedinica sa zajedničkog čipa. Sveti čin spajanja bi prvi put trebalo da bude učinjen u i866 procesoru (i786, odnosno Micro 2000, već smo opisali prošla godina). Pored i586 koji konstantno kasni, i860XP, novi član 860 porodice, generalna je proba nova strategije.

LIČNA KARTA 860 XP

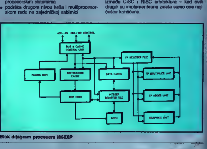
"Intelova" specifikacija kaže: i860XP je 860 procesor druge generacije koji krase sledeće odlika:

- 0,8-mikrometarska CMOS tehnologija sa 2,5 miliona tranzistora
- 32-bitni RISC celobrojni procesor sa izvršavanjem najvećeg dela naredbi u jednom taktu i 32 32-bitna registra opšta namene
- 64-bitni vektorski FP procesor sa odvojenim sabiračem i množačem koji rade paralelno i 32 32-bitna FP registra
- 64-bitni 3-D grafički procesor
- stranična MMU sa kešom za 64 4-kilobajtne i 16 4-megabajtnih stranica
- 16 kilobajtni keš za naredbe 64-bitne širine
- 16 kilobajtni keš za podatka 128-bitne širine
- 64-bitna unutrašnja sabirnica za naredbe brzine 400 Mb/s
- 128-bitna unutrašnja sabirnica podataka brzine 800 Mb/s
- DCCU jedinica za kompatibilnost sa multiprocesorskim sistemima
- podrška drugom nivou keša i multiprocesorskom radu na zajedničkoj sabirnici
- 64-bitna nemultipleksirana spoljna sabirnica maksimalna brzine 400 megabajta/s na 50 MHz
- 262-pinsko PGA kućište u matrici 19 x 19
- frekvencije 40 i 50 (uskoro i 60) MHz

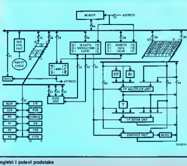
Pogledajmo detaljnije arhitekturu nove „Intelove" uzdanice, koja je inače dobrim delom koncipirana po ugledu na „Cray" superračunare. Veliki broj internih registara, nekoliko širokih sabirnica koje povezuju unutrašnje jedinice, omogućujući time veoma veliku internu brzinu prenosa i paralelan rad tih jedinica, ultrabrza spoljna sabirnica sa opcionom keš-memorijom drugog nivoa i visoko efikasnim multiprocesorskim radom daju performanse koje do sada nisu mogle biti zamišljene na nivou mikroračunara, uz istovremenu potpunu koegzistenciju sa 486/586 procesorima na istoj sabirnici.

CELOBROJNI PROCESOR

Mada mu nije baš najjača strana, celobrojni deo 860XP je ipak srce procesora. On nadgleda celokupan rad 860XP, izvršava **load**, **store**, celobrojne, ulazno/izlazne i kontrolne operacije, kao i prihvatne naredbe za FP i grafičku jedinicu. Kao i ostali RISC procesori, i 860XP celobrojna jedinica sve prenose između registara, memorije i I/O obavlja isključivo preko **Load** i **Store** naredbi. Raspolaža sa 32 32-bitna registra opšte namene i 12 kontrolnih registara vezanih za ceo 860XP. Od 87 860XP naredbi, 43 se u potpunosti izvršavaju u celobrojnoj jedinici, a sve ostale se nadgledaju odatle. Najveći deo naredbi 860XP je troadresni, u skladu sa njegovom troadresnom arhitekturom. U listi naredbi se jasno može videti ogromna razlika između CISC i RISC arhitektura – kod ovih drugih su implementirane zaista samo one najčešće korišćene.



Blok dijagram procesora i860XP



Registri i putevi podataka

Tipovi podataka s kojima celobrojna jedinica 860XP radi su približno jednaki kao i oni kod 386/486 8, 16 32 64-bitni označeni i 32- i 64-bitni neoznačeni celi brojevi Ovde nema dugih bit i bajt stringova koji su kao tip podataka prisutni na 386/486 arhitekturi

860XP celobrojne jedinice ne koristi najnovije tehnike superskalarnog ili supercevovodnog (superpipelining) izvršavanja naredbi koje omogućavaju izvršavanje i više naredbi u jednom taktu, kakve inače koriste procesori tipa Sun/Texas SuperSPARC MIPS R4000, MC 88110 IBM POWER, Inmos T 9000 i nekoliko drugih Međutim pošto je preko 90% praktičnih implementacija 860 procesora vezano za računanje u pokretnom zarezu i 3-D grafike to superskalarnost supercevovodi i čiste 64-bitnost nisu smatrani neophodnim u ovom trenutku za celobrojnu jedinicu pošto one manje više ionako služi isključivo kao pomoćnik FP i grafičkoj jedinici To ne znači da je isključena mogućnost nadograđivanja na superskalarnost ili punu 64-bitnost Naprotiv u startu je zamišljeno da 860 porodica svoju punu zrelost dostigne u 64-bitnom svetu Ono što jeste bilo vazno prilikom dizajniranja je da FP jedinica bude maksimalno brza jer ona je osovina uspeha 860 porodice

VEKTORSKI FP PROCESOR

860 porodica u svom FP delu implementira vektorsku cevovodnu arhitekturu, koja je u svetu mikroračunara prisutna već skoro jednu deceniju preko specijalizovanih vektorskih FP procesora. Kakva je razlika između klasičnih (skalarnih) i vektorskih FP procesora?

Skalarni procesori za rad u pokretnom zarezu rade na principu da tokom izvršenja programa sledeća naredba ne može biti započeta dok ona čije je izvršavanje u toku ne bude okončana – to jest dok ne da rezultat Na taj način brzina izvršavanja biva neposredno ograničena brzinom izvršavanja pojedinih naredbi Vektorski procesori s druge strane imaju cevovodni (pipeline) način izvršavanja kod koga prethodno ograničenje ne važi Kod pipeline izvršavanje zahvaljujući podeli izvršenja naredbe na onoliko delova koliko ciklusa traje apsolutno vreme izvršenja (kod svih FP naredbi 860XP sem 64-bitnog množenja su to 3 ciklusa pa samim tim i 3 nivoa cevovoda), po jedna naredba može biti startovana svakog ciklusa Takođe u svakom ciklusu tada dobijamo jedan rezultat. To se postiže tako što tri nivoa cevovoda paralelno rade, samim tim i tri uzastopne naredbe u različitim fazama izvršenja se paralelno obrađuju

FP procesor u 860XP poseduje mogućnost i skalarnog i vektorskog rada Kao i stari 860 i većina novih RISC procesora i 860XP ima odvojene paralelne FP sabirač i FP množač koji mogu raditi u 32-bitnoj i 64-bitnoj preciznosti Kako oni imaju odvojene sabirnice a 860 ima „dual operation" naredbe koje paralelno koriste obe jedinice to je 860XP u stanju da maksimalno obradi 100 miliona FP operacija u sekundi (100 MFLOPS) na 50 MHz „Dual operation" rad, koji je inače izuzetno koristan za mnoge česte poslove kao rešavanje sistema jednačina ili brze Furijeove transformacije (FFT) pomažu i tri posebna registra KR, KI i T, koji smeštaju rezultate jedne dual-op naredba i koriste ih kao ulazne podatke za sledeću dual-op naredbu Kako se 64-bitno množenje izvršava u 4 ciklusa to se na svaka 3 ciklusa uzastopnih rezultata mora dodati još jedan ciklus čekanja prilikom 64-bitnog rada FP množača

Mogućnost za dodatnu paralelizaciju daje „dual instruction" način rada gde se paralelno učitavaju i izvršavaju jedna celobrojna i jedna FP naredba Pri tome je u 64-bitnoj reči koja se učitava u procesor FP naredba u donja 32 bita a celobrojna naredba u gornja 32 bita Pošto FP naredba opet može biti „dual operation" znači da je moguće da u svakom takt-ciklusu izvršimo tri operacije jednu celobrojnu i dve u pokretnom zarezu

3-D GRAFIKA

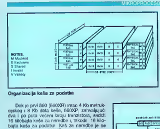
Usko vezan za FP procesor je 64-bitni grafički deo 860XP CPU Grafička jedinica podržava 3-D crtanje neposredno u VRAM memoriji slike sa skrivanjem nevidljivih površina i senčenjem preko brzog Z-Buffer algoritma koji koristi deo programske memorije za smeštanje Z koordinata tačaka koje će biti obrađivane 860XP inače podržava 8-bitne 16-bitne i 32-bitne tačke (pixel) kao tipove podataka, pri čemu je 8-bitni piksel bez unutrašnje podele, 16-bitni piksel sa podelom 6-6-4 (to, na žalost, onemogućava upotrebu Hi-Color tipa sa 32 768 boja čija organizacija je 5-5-5-1), a 32-bitni piksel ima organizaciju 8-8-8-8 – po 8 bita za svaku osnovnu boju i 8 bita za transparentnost „overlay" ili tekst ravan što daje već poznatih svih 16.8 miliona boja istovremeno dostupnih Nezavisno od veličine piksela grafička jedinica uvek odjednom uzima 64 bita Elementarne 3-D grafičke naredbe koje su implementirane su odlična osnova za ultrabrzo „Gouraud" i „Phong" senčenje na primer Veoma velika brzina 860XP procesora u 3-D grafici (do 1 000 000 transformisanih 3-D vektora i 50 000 senčenih trouglova u sekundi) je rezultat timskog rada FP i grafičke jedinica uz brzinu širokih unutrašnjih i spoljnih sabirnica.

MMU I KEŠ MEMORIJE

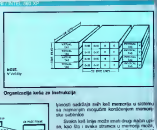
"Intelov" 860XP kao i drugi mikroprocesori opšte namene adresira memoriju u bajtovima, u ovom slučaju unutar 32-bitnog straničenog adresnog prostora od 4 Gb 860XP normalno radi u „little endian" (najniži bit na najnižoj adresi – kao 486) radu s tim što u supervizorskom načinu poseduje i mogućnost „big endian" (najniži bit na najvišoj adresi – kao MC 68030 recimo) rada

MMU mehanizam kada je uključen mapira virtuelne adrese u fizičke pre pristupa memoriji Adresna translacija u 860XP je kompatibilna sa onom u MMU jedinicama 386 i 486 (takođe i 586) procesora i obezbeđuje ovom procesoru osnovne sposobnosti potrebne za višeposlovne i višekorisničke sisteme sa virtuelnom memorijom i zaštitom na nivou stranica MMU u 860XP za razliku od većine drugih MMU koje koriste samo podelu na stranice od 4 kilobajta, omogućava i podelu na 4 Mb stranice što je pogodno za grafičke i druge aplikacije koje pristupaju velikim blokovima memorije Naime, pošto proces translacije adresa preko kataloga stranica u memoriji zna da potraje dugo svaka MMU ima poseban ATC (keš za translaciju adresa) u koji smešta izvestan broj adresa najčešće korišćenih stranica. Naprimer, 80486 ima ATC za 32 4 Kb stranice 68040 za 2 x 64 4 Kb stranice (odvojeno za naredbe i za podatke), a 860XP ima ATC za 64 4 Kb i 16 4 Mb stranica Pošto kod aplikacija koje pristupaju velikim delovima memorije nijedan ATC za 4 Kb stranice ne bi zadovoljio uvođenje stranica od 4 megabajta rešava taj problem

S obzirom da spoljna memorija kakva god bila ne može raditi bez stanja čekanja sa 50 MHz procesorima i njihovim „burst" načinima pristupa potreba za što većim efikasnijim i bržim unutrašnjim kešom je kod ovakvih procesora nezaobilazna. Uz to brz inteligentan i 16 ili više puta veći keš drugog nivoa bi pored dodatnog ubrzanja rada usled manjeg broja pristupa DRAM obezbedio i još jednu korisnu prednost – manju zauzetost sistemske sabirnice i time manju gužvu kod multiprocesorskih sistema, što opet daje porast performansi bliži linearnom kod dodavanja više procesora. 860XP implementira jedan ili dva nivoa keša.



Organizacija keša za podatke



Organizacija keša za instrukcije

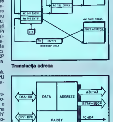
Dok je prvi 860 (860XR) imao 4 Kb instrukcijskog i 8 Kb data keša, 860XP, zahvaljujući dva i po puta većem broju tranzistora, sadrži 16 kilobajta keša za naredbe i, takođe 16 kilobajta keša za podatke. Keš za naredbe je sa ostalim delovima 860XP povezan 64-bitnom sabirnicom provodnosti 400 megabajta u sekundi, a 128-bitna veza keša za podatke sa ostatkom procesora u stanju je da prebaci celih 800 megabajta/s. Oba keša, sa procentima pogotka između 90% i 96%, koriste efikasnu četvorostruko set asocijativnu organizaciju. LRU (Least Recently Used) algoritam se koristi za održavanje prisustva najčešće korišćenih informacija u kešu, a MESI (Modified, Exclusive, Shared, Invalid) protokol za održavanja valjanosti informacija u kešu u sistemima sa više 860XP na zajedničkoj memorijskoj sabirnici. MESI protokol je bio neophodan da bi 860XP keš pored ostalog imao izbor načina pisanja write-through, write-back ili write-once.

Kao što smo u prethodnim brojevima pisali, postoji nekoliko načina upisa podataka iz CPU u RAM preko keša ukoliko je linija sa odgovarajućom adresom prisutna u kešu.

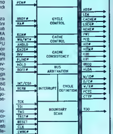
"Write-through" način obnavlja podatke istovremeno i u kešu i u memoriji, s tim što procesor nastavlja rad tek kad je upis završen i u memoriji. Bez obzira što u performansama zaostaje za ostalim načinima, „Write-through" način obavezan je kod, na primer, meduprocesorskih poruka ili upisa u video RAM, jer ono što se upisuje treba odmah potom i da se vidi na ekranu.

"Write-back" način, koji je default za 860XP, upis vrši samo u keš, dok se konačni upis u RAM vrši tek kada u keš treba smeštati druge podatke, kada drugi procesor ili DMA upravljač traži te promenjene podatke i kada se keš čisti. Ovaj način ubrzava rad jer CPU ne čeka na memoriju i dodatno smanjuje eventualnu gužvu na zajedničkoj memorijskoj sabirnici višeprocesorskog sistema, međutim zahteva znatno složeniju i inteligentniju upravljačku logiku da ne bi došlo do eventualnih problema u izvršenju programa. Iz tog razloga 486, recimo, nema „write-back" mogućnost rada svog unutrašnjeg keša, a i čip setovi za 486 retko koriste „write-back" rad spoljnog keša. Tek najnovija generacija čip setova za ovu godinu uvela je ovaj način rada maltene kao obavezu.

Pored „buffered write through" koji nije korišćen kod 860XP, postoji još jedna kombinacija prethodna dva osnovna načina – „write-once". Kod ovog načina „write-through" se koristi za prvi upis u keš liniju, a „write-back" za sledeće. Ovaj način se za unutrašnji keš 860XP koristi kod sistema sa kešom drugog nivoa i kod višeprocesorskih sistema (onaj prvi upis obaveštava keš drugog nivoa i druge procesore da je linija promenjena) radi održavanja valjanosti sadržaja svih keš memorija u sistemu sa najmanjim mogućim korišćenjem memorijske sabirnice.



Translacija adresa



Kontrolni signali

Svaka keš linija može imati drugi način upisa, kao što i svaka stranica u memoriji može, pored zabrane keširanja, imati izbor jedne od tri navedene tehnike upisa. Šta je, inače, keš linija?

Pošto sve keš memorije, pa tako i ove u 860XP, rade na „locality of reference" principu, znači da se smatra da je, ako se naredbi ili podatku na nekoj adresi pristupa u određenom trenutku, velika verovatnoća da će se uskoro pristupati i susednim lokacijama. Iz tog razloga, da bi se neki od tih podataka unapred učitali koristeći prednosti brzog „burst" načina rada spoljnih sabirnica, učitavanje u keš se ne vrši reč po reč, nego se učitava cela linija sa 4 uzastopne reči (32 bajta). Tako je svaka od unutrašnjih keš memorija 860XP podeljena na po 512 32-bajtnih linija, gde svaka ima svoju logičku (virtuelnu) i fizičku adresu, kao i MESI status u posebnoj tag memoriji. Prilikom „burst" načina učitavanja, prva reč se učitava u onoliko taktova koliko memorija zahteva (2 ako je učitavanje iz spoljnog keša), a ostale 3 reči iz linije u samo po jednom taktu. Da bi to zaista i radilo tako, RAM mora koristiti brzi stranični ili pak način statičkih kolona i, naravno, bar dvostruko preplitanje. Sa 2-1-1-1 ciklusima dobijamo praktičnu maksimalnu brzinu spoljne sabirnice od 320 megabajta/s.

Za višeprocesorske i uopšte sve 860XP sisteme koji ciljaju ka najvišim performansama, „Intel" nudi i visokointeligentni sistem keša drugog nivoa, sastavljen iz 82495XP keš upravljača i 82490XP 32-kilobajtnih keš čipova. Ovaj keš sistem se vezuje neposredno na 860XP, praktično ga izolujući od spoljnog sveta i omogućujući primenu čak i sporih memorijskih interfejsa bez značajnog gubitka na brzini. Kao što smo rekli, glavna primena inteligentnih sekundarnih keš memorija su ipak višeprocesorski sistemi, gde one mogu bitno doprineti ukupnim performansama jer, sa procentom pogotka preko 99%, gotovo u potpunosti eliminišu mogućnost istovremenih pristupa memoriji dva ili više procesora.

82495XP keš upravljač podržava do 18 82490XP RAM kola – 512 Kb keša sa paritetom. Dvostruko set-asocijativna organizacija, MESI prootokol, dužina linije jednaka, dvostruko ili četvorostruko veća od one u 860XP kao i 64-bitna ili 128-bitna širina memorijske sabirnice podataka dovoljno govore o konfigurabilnosti. 82495XP/82490XP podsistem omogućava izbor iz ista tri načina upisa kao i 80860XP unutrašnji keš, kao i paralelni dual-port rad na 860XP sabirnici, s jedne, i memorijskoj sabirnici, s druge strane. Memorijski interfejs, koji je zavisan od dizajnera same ploče, može koristiti adresni cevovod, stranični način rada DRAM,

Naredbe 860XP

Naredba	Opis
Load/Store	
ld.x	Load integer
st.x	Store integer
fld.y	FP load
fst.y	FP store
pfld.y	Pipelined FP load
pst.d	Pixel Store
reg/reg move	
ixfr	transfer integer to FP register
fxfr	transfer FP to integer register
Integer Arithmetic	
addu	Add unsigned
adds	Add signed
subu	Subtract unsigned
subs	Subtract signed
Shift	
shl	Shift left
shr	Shift right
shra	Shift right arithmetic
shrd	Shift right double
Logical	
and	Logical AND
andh	Logical AND high
andnot	Logical AND NOT
andnoth	Logical AND NOT high
or	Logical OR
orh	Logical OR high
xor	Logical XOR
xorh	Logical XOR high
Control-Transfer	
br	Branch direct
bri	Branch indirect
bc	Branch on CC
bc.t	Branch on CC taken
bnc	Branch on not CC
bnc.t	Branch on not CC taken
bte	Branch if equal
btne	Branch if not equal
bla	Branch on LCC and add
call	Subroutine call
calli	Indirect Subroutine call
intovr	Software trap on integer overflow
trap	Software trap
FP Multiplier	
fmul.p	FP multiply
pfmul.p	Pipelined FP multiply
pfmul3.dd	Stage pfmul
fmlow.p	FP multiply low
frcp.p	FP reciprocal
fsqr.p	FP square root
FP Adder	
fadd.p	FP add
pfadd.p	Pipelined FP add
famov.r	FP adder move
pfamov.r	Pipelined FP adder move
fsub.p	FP subtract
pfsub.p	Pipelined FP subtract
pfgt.p	Pipelined greater than compare
pfeq.p	Pipelined equal compare
fix.v	FP -> int conversion
pfix.v	Pipelined fix
ftrunc.v	FP -> truncation
Dual Operation	
pfam.p	Pipelined FP add and mul
pfsm.p	Pipelined FP sub and mul
pfmam.p	Pipelined FP mul with add
pfmsm.p	Pipelined FP mul with sub
Long Integer	
fisub.z	Long-int subtract
pfisub.z	Pipelined fisub
fiadd.z	Long-integer add
pfiadd.z	Pipelined fiadd
Graphics	
fzchks	16-bit Z-buffer check
pfzchks	Pipelined fzchks
fzchkl	32-bit Z buffer check
pfzchkl	Pipelined fzchkl
faddp	Add with pixel merge
pfaddp	Pipelined faddp
faddz	Add with Z merge
pfaddz	Pipelined faddz
form	OR with MERGE register
pform	Pipelined form
I/O	
ldio.x	Load I/O
stio.x	Store I/O
ldint.x	Load interrupt vector
System Control	
flush	Cache flush
ld.c	Load from control reg
st.c	Store to control reg
lock	Begin interlocked sequence
unlock	End interlocked sequence
scyc.x	Special bus cycles
Assembler Pseudo-Ops	
mov	Integer move
fmov.r	FP reg reg move
pfmov.r	Pipelined fmove
nop	Core no-op
fnop	FP no-op
pfle.p	Pipelined FP less-than-or equal

preplitanje i druge tehnike ubrzanog pristupa. Uostalom, da li ste ikada videli RAM čip pa makar i statički, sa 84 pina?

Uzgred, 82495XP/82490XP sistem je kompatibilan sa 82495DX/82490DX sistemom za 50 MHz 486.

Pogledajmo primer jedne konfiguracije. Na 860XP je vezan keš sistem sa 512 Kb keša 128-bitnom memorijskom sabirnicom i keš linijama od 64 bajta (4 x 128 bita). Ako koristimo četvorostruko preplitanje 128-bitnih banaka 60 ns DRAM, praktična brzina učitavanja u „burst" načinu bila bi veća od brzine 860XP spoljne sabirnice!

Spoljna sabirnica 860XP, inače, podržava paritet, vodove za „Boundary scan" testiranje ispravnosti procesora i arbitraciju između više procesora. Burst ciklusi na njoj, kao i na 82495XP/82490XP sabirnici, mogu obuhvatiti i više od četiri prenosa.

VIŠE PROCESORA ZAJEDNO

Kad što smo videli, 860XP i sekundarni keš su u startu prilagođeni višeprocesorskom radu. Sa svojim sekundarnim keš memorijama i dobrom sabirnicom, do 8 860XP i još poneki 486 ili 586 mogu lepo koegzistirati zajedno, uz veoma veliku brzinu obrade paralelabilnih poslova. Međutim, tu je potrebno još nešto. To nešto je CCU (Concurrency Control Unit), jedinica koja u višeprocesorskom sistemu omogućava procesorima da zajedno izvršavaju program. CCU je poseban Intelov VLSI čip, čiji su registri memorijski mapirani u dve 4 Kb stranice i koji sinhronizuje paralelan rad više procesora na istom problemu. Da bi se paralelna arhitektura iskoristila, potrebno je da programi budu kompajlirani paralelizujućim vektorskim kompajlerima koji generišu naredbe za pristup CCU i ponašaju se po „Intelovoj" PAX (Parallel Architecture eXtensions) normi. Da bi se takvi programi mogli izvršavati i na sistemima sa samo jednim 860XP, u njega je ugrađena DCCU (Detached CCU) koja oponaše rad CCU onda kada je potrebno da se izvrši neki program napravljen za paralelni sistem. Na primer, sistem sa 4 860XP, svaki sa po 512 Kb keša, bio bi u stanju da izvuče skoro 400 MFPLOPs za cenu koja ne bi morala da premaši 30 000 USD za podsistem sa 64 Mb RAM i nekom grafikom visoke rezolucije u 16 miliona boja. A sa takvim sistemom moguće je praviti animaciju u realnom vremenu. Nameće se poređenje sa „Inmos" T 9000 transpjuterom, koji se na tržištu pojavio otprilike u isto vreme kada i 860XP. T 9000, kao što znamo, podržava i ogromne višeprocesorske sisteme sa brzim 100 Mbit/s linkovima, gde svaki procesor ima svoj RAM. T 9000 je, uz to, nešto brži u celobrojnim poslovima zbog superskalarnosti, ali i sporiji od 860XP u matematici, jer nema vektorski rad.

Inače, kako se saznaje, 860XP bi trebalo da može da neposredno koegzistira na istoj memorijskoj sabirnici sa novim 586. Kada se bude znalo kako će to stvarno izgledati, moći će se nešto više reći.

„Intelov" 860XP sigurno nije najsavršeniji procesor koji se u ovom trenutku može naći, ali je u samom vrhu. Performansa od preko 40 integer SPECmarks, 48 VAX MIPS, 21 Linpack MFLOPS i izuzetno brza 3-D grafika dovoljno govore. Solidna prihvaćenost prve generacije 860, sve bolja softverska podrška i u svetu PC (brojne matematičke, CAD i 3-D grafičke aplikacije će imati ili već imaju 860 verzije), uz lake mogućnosti dodatne optimizacije za novi procesor (jedan ili više paralelnih) vezu sa 486/586 i samo „Intelovo" ime daje garancije da 860 porodica čvrstim korakom gazi dalje.

ŠTA IMA NOVO

Procesori

NAJBRŽI NA SVETU

Prošla, 1991. godina bila je u svetu mikroprocesora označena i najavom velike borbe za vladavinu novom generacijom mikroračunara. U borbi su ACE konzorcijum sa MIPS R4000, SPARC grupa sa SuperSPARC, IBM/"Apple"/"Motorola" sa POWER, HP sa „Precision" i naravno „Intel" sa svojim arhitekturama. Početak ove godine doneo je novog takmaca kome se malo ko nadao – DEC ALPHA.

"Digital Equipment", tvorac VAX arhitekture koja je bliži ili dalji koren svim današnjim savremenim CISC procesorima, odlučio je da ozbiljno uđe u RISC trku sa svojom porodicom „Alpha", koja bi po fleksibilnosti, skalabilnosti i performansama trebalo da nadmaši sve konkurente i da bude jedna od glavnih u iduće dve decenije. „Alpha" arhitektura je potpuno 64-bitna, uključujući puno 64-bitno adresiranje (2^64 bajta adresnog prostora = preko 18 triliona bajta), sa barem 128-bitnim unutrašnjim i 64-bitnim spoljim data sabirnicama, superskalarnim celobrojnim, vektorskim FP i brzim 2-D i 3-D grafičkim jedinicama, velikim keš memorijama u dva nivoa i rasponom taktova od 50 do 200 MHz (100 do 400 MIPS) u prvoj seriji (do sredine 1993.). Prvi 150 MHz prototip primerci su već istestirani u gotovim konfiguracijama („Olivetti" ima prototip), a proizvodni primerci se očekuju na jesen u većim količinama.

Usled kraha MIPS-a i njegovog prelaska u vlasništvo „Silicon Graphics", ACE inicijativa je prilično uzdrmana. Nuđenjem procesora koji je višestruko brži od R4000, uz prebacivanje Open VMS, OSF Unix i forsiranje kompatibilnosti sa „Intel", VAX i MIPS/ACE platformama, DEC je opasno ugrozio proizvođače MIPS procesora.

Inače, prvi mušterija je „Cray Research", koji će koristiti „Alphu" u svom novom MPP (Massive Parallel Processing) – visokoparalelnom superkompjuteru sa nekoliko stotina „Alphi" u paraleli. I neki od velikih tajvanskih PC proizvođača su zainteresovani za Alphu.

DEC namerava da sa „Alpha" arhitekturom nadmaši konkurente i da ove arhitekture bude jedna od glavnih u iduće dve decenije. „Alpha" arhitektura, bez sumnje, za to ima velike šanse. Opširniji prikaz očekujte u jednom od narednih brojeva.

Fontovi

BITSTREAM I TRUETYPE

„Microsoft" nije jedina kompanija koja nudi *TrueType* fontove za *Windows* – poznati proizvođač fontova „Bitstream" je pustio u prodaju novi *TrueType Font Pack* koji sadrži 40 fontove po preporučenoj ceni od $79. „Microsoft" sa druge strane nudi sopstveni set fontova za $50, ali „Bitstream" tvrdi da njihovo rešenje sadrži fontove koji se nigde drugde ne mogu nabaviti. Registrovani korisnici *Bitstream Typeface* proizvoda mogu da dobiju paket za samo $30, a kompanija je uporedo sa tim objavila da planira početak prodaje seta od preko 1 000 PostScript Type 1 fontova tokom juna ove godine.

Kontakt adresa: Bitstream Inc., 215 First St. Cambridge, MA 02142 (617) 497-6222

OD ČVORA DO ČVORA

Interaktivne poruke, koje smo u prethodnom nastavku obradili, nisu pogodno sredstvo za slanje većih količina podataka. Taj posao se može uspešno obaviti pomoću sledeće standardne usluge BITNET-a, poznate pod imenom *File transfer* ili prenos datoteka.

Zvonko Oršolić

Glavna odlika usluge prenosa datoteka kroz BITNET je njegova nesimetričnost. Datoteku možete iskopirati sa lokalnog čvora, tj. sa čvora na kome radite, na bilo koju BITNET mašinu (mašina=računar=čvor). Ne postoji, međutim, mogućnost da date komandu kojom biste datoteku koja se nalazi na udaljenom računaru iskopirali na lokalni računar! To nije moguće čak ni kada imate otvorene radne naloge, tj. pravo pristupa, na obe od mašina među kojima se vrši transfer.

Prenos datoteka je, poput prenosa interaktivnih poruka, usluga specifična za BITNET. Moguć je samo ukoliko su oba učesnika, i računar sa koga se šalje poruka i računar na koga se šalje datoteka, članovi BITNET-a. S obzirom na veličinu BITNET-a, korišćenjem ovog tipa komunikacija dostupno je oko 3500 velikih računara, tj. više stotina hiljada ljudi širom sveta. U svetlu današnjih mreža, to je cifra koja nije ni prevelika ni premalena. Videćemo, međutim, da se ova mrežna usluga retko koristi za komunikacije među ljudima, a mnogo češće za komunikaciju sa mrežnim serverima.

FAZE PRENOSA

Prenos datoteke u BITNET-u odvija se u više faza:

1. Korisnik daje komandu za prenos datoteke. Po davanju komande, njegov posao je završen i on se može baviti nekim drugim aktivnostima. Kopija njegove datoteke smešta se u područje u kome se čuvaju datoteke koje treba postati prema odredišnom računaru. Tu datoteka čeka da link kojim treba da izađe u svet postane dostupan i slobodan. Drugim rečima, datoteka staje u red sa svim drugim datotekama koje čekaju da izađu napolje i strpljivo čeka da na nju dođe red.
2. Datoteka se prenosi od čvora do čvora, duž putanje kojom se stiže do odredišta. Prenošenje datoteke od čvora do čvora prikazano je na slici 1.
3. Datoteka stiže na odredišni računar. Prijemni softver analizira zaglavlje datoteke i, u skladu sa tim, preduzima određene akcije. Tipična akcija koja se preduzima kada su u pitanju „obične" datoteke je da se upravo pristigla datoteka izvadi iz ulaznog reda i smesti u područje na disku koje ćemo nadalje zvati centralno prijemno područje (*network file directory, receive buffer, receive catalog*).
4. Korisnik se prijavljuje na računar i dobija poruku da se u centralnom prijemnom području nalazi jedna ili više datoteka koje su pristigle na njegovu adresu. Korisnik daje komande za preuzimanje datoteke iz centralnog prijemnog područja i datoteku premešta u svoj katalog.

S obzirom na prirodu usluge prenosa datoteka u BITNET-u, učešće korisnika svodi se na dve aktivnosti: jednu kada daje zahtev da se datoteka prenese i drugu kada iz centralnog prijemnog područja mora da preuzme datoteku i prenese je u svoj katalog. Prvu od ovih aktivnosti zvaćemo slanje, a drugu prijem.

Prikažimo sada gore opisanu proceduru nešto detaljnije. Ne gubeći previše na opšta mesta, ponovo se ograničavamo na samo jedan operativni sistem. I ovoga puta, kao i u prethodnom nastavku, izbor je VAX/VMS sa JNET softverom u ulozi NJE podrške, dakle mašina kao što je UBBG. Tu se za slanje datoteka koristi komanda SEND/FILE, a za prijem datoteka uslužni program RECEIVE.

SLANJE DATOTEKE

Osnovni oblik komande za slanje datoteka kroz mrežu je:

```
$ SEND/FILE -
  naziv_datoteke -
  username@nodename
```

gde umesto **naziv_datoteke** treba staviti naziv datoteke u nekom od standardnih VAX/VMS oblika (recimo, **ime.tip** ili **ime.tip;verzija**), umesto **username** naziv korisničkog naloga kome se datoteka šalje, a umesto **nodename** ime čvora na kome taj korisnik radi.

PRENOS DATOTEKE

Putovanje datoteka kroz mrežu prikazano je na slici 1. Svojstvo BITNET-a je, kao što smo i ranije govorili, da se datoteke prenose u celini. Na putu od čvora do čvora uvek se prvo prenese kompletna datoteka, a tek onda se na prijemnom čvoru ispituje prema kom čvoru je dalje treba slati. Pogledajmo detalje na jednom primeru.

Neka je korisnik PERA sa čvora AAA poslao datoteku koja sadrži proizvoljan tekst i zove se ZA.PRENOS korisniku MIKA na mašini CCC. Između mašina AAA i CCC nema direktne veze, ali se komunikacije među njima mogu ostvariti posredstvom mašine BBB.

Davanjem komande

```
$ SEND/FILE ZA.PRENOS MIKA@CCC
```

Pera je obavio svoj posao. To što će datoteka do odredišta na drugom kraju sveta putovati možda i više dana, stvar je mreže, a ne pošiljaoca. Na mašini AAA datoteka ZA.PRENOS se pakuje u mrežni oblik. To pakovanje obuhvata konverziju iz ASCII skupa znakova u EBCDIC, kao i pakovanje zapisa u formate pogodne za prenos kroz mrežu. Tako spakovanoj

PRAVILA I ZABRANE

EARN/BITNET je mreža namenjena za razmenu informacija, a u skladu sa akademskom, obrazovnom i istraživačkom prirodom organizacija članica mreže.

Mreža je otvorena za studente u meri u kojoj je to i organizacije (fakultet) kojoj pripadaju odluči.

Zabranjene su: upotreba mreže u komercijalne svrhe, upotreba mreže za političke i religiozne aktivnosti, upotreba ružnih reči, uvreda i psovki, distribucija licenciranog softvera, masovno slanje interaktivnih poruka, datoteka i elektronske pošte slučajno izabranim nepoznatim ljudima, neautorizovani pristup udaljenim mašinama u cilju proveravanja, podmetanja virusa i trojanskih konja i slično.

U cilju očuvanja performansi mreže zabranjuje se prenos velikih datoteka (više megabajta) preko mreže. Velike datoteke treba razbiti u više manjih i prenositi u delovima.

U slučaju komunikacija sa drugim mrežema posredstvom EARN/BITNET-a, moraju se poštovati i pravila koja važe za te mreže.

Ko može i kako?

Institucije koje spadaju u klasu akademskih, obrazovnih i istraživačkih institucija, a i mnoge druge uz posebnu saglasnost, mogu da postanu članice organizacije EARN – Jugoslavija. Odluke i preporuke o prijemu u članstvo donose organi EARN – Jugoslavija (skupština, direktor, upravni i tehnički odbor). Informacije o prijemu u članstvo, formulari koje treba popuniti, pravila i obaveze koje treba prihvatiti, sve se to može dobiti u Rektoratu Beogradskog univerziteta. Telefon na koji se možete obratiti je 011-636-941, a odgovorna osoba je g. Rade Jovančević.

Članice EARN-a imaju pravo da svoje računare uključe u BITNET.

Zaposleni u institucijama koje su članice EARN-a imaju i pravo da podnesu zahtev za otvaranje naloga na nekoj od postojećih BITNET mašina (za sad samo YUBGSS21 i YUBGEF51). Ukoliko postoji mogućnost za otvaranje naloga, oni će obično biti i otvoreni. Trenutno, oko 1500 ljudi ima naloge za elektronsku poštu na ove dve mašine.

Posredstvom mrežnog prolaza realizovanog na mašini YUBGEF51, još oko 6000 ljudi širom Srbije ima mogućnost da na svojim mašinama koristi usluge BITNET-a.

Odakle se može pristupiti

Trenutno imamo samo dve mašine registrovane u BITNET-u. Jedino je na ovim mašinama moguće koristiti potpuni skup BITNET usluga (elektronska pošta, prenos datoteka, prenos interaktivnih poruka). To su mašine u Republičkom zavodu za statistiku Srbije i na Elektrotehničkom fakultetu u Beogradu.

Računar YUBGEF51 (UBBG) na Elektrotehničkom fakultetu obavlja, pored dosta drugih stvari, i ulogu mrežnog prolaza za elektronsku poštu između BITNET-a i DECnet-a SNTIS-a. Zahvaljujući tome, korišćenje usluga BITNET-a (istina samo elektronske pošte) moguće je i sa VAX/VMS i UNIX mašina koje se nalaze u sledećim ustanovama: Elektrotehnički fakultet Beograd (ostale mašine), Građevinski fakultet Beograd, Fakultet organizacionih nauka Beograd, Univerzitet Novi Sad, Narodna biblioteka Srbije Beograd, Jugoslovenski bibliografski institut Beograd, Biblioteka Matice Srpske Novi Sad, Gradska biblioteka Subotica, Gradska biblioteka Sombor i Nacionalna i univerzitetska biblioteka Priština.

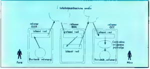
Na nekim od navedenih lokacija rad je moguć samo iz prostorija institucije, dok nekoliko lokacija (recimo Univerziteti u Beogradu i Novom Sadu) imaju i mogućnost modemskog pristupa sa udaljenih personalnih računara.

Dalji razvoj Bitnet-a

U Beogradu je toku puštanje u rad novog univerzitetskog računara koji bi, čim dođe u redovno radno stanje, trebalo da preuzme ulogu nacionalnog BITNET čvora.

Nova moderna univerzitetska mreža u Beogradu i Srbiji, čije izgradnje upravo započinje, znatno

Slika 1



Prenošenje datoteke od čvora do čvora

Slika 2

```
$ RECEIVE
Files received for ZVONKO
Source file          Class      Node User          Date Time  Seconds
POSTGRES ME 84       PUN A ANTWUM11 TRICKLE  17 Apr 1992 17 26    456
VMSDEM2 MAR 57       PUN A ANTWUM11 TRICKLE  17 Apr 1992 13 06    225
VMSHEX2 MAR,56       PUN A ANTWUM11 TRICKLE  17 Apr 1992 11 27    218
LZCOMPRE SHARE,66    PUN A ANTWUM11 TRICKLE  13 Apr 1992 23 23   2022
VMS TCPIP,53         PUN A  IMIPOLI TRICKLE  13 Apr 1992 17 46     41
FILES 6648014,52     PUN A     PUCC BITFTP1   7 Apr 1992 11 37   5027
FILES 9848013,51     PUN A     PUCC BITFTP1   7 Apr 1992 11 36   6027
FILES 9848012,50     PUN A     PUCC BITFTP1   7 Apr 1992 11 36   6020
RZRZ9107 TLB 46      PUN A ANTWUM11 TRICKLE   6 Apr 1992 09 36   2279
FILES BITCTRL,47     PUN A     PUCC BITFTP1   7 Apr 1992 11 37      8
EARN 011 README 46   PUN A YUBGEF51 ZVONKO    7 Apr 1992 15 06    468
RECEIVE>
```

Pokretanje uslužnog programa RECEIVE i informacije koje on ispisuje

datoteci dodaje se još i zaglavlje u kome piše kome se datoteka šalje, kada je poslata, ko je šalje i još nekoliko informacija.

Ovako pripremljena datoteka smešta se na čvoru AAA u izlazni red čekanja prema čvoru BBB, dakle u izlazni red prema čvoru kroz koji ona mora da prođe da bi stigla do čvora CCC.

Po prijemu na čvoru BBB se analizira zaglavlje datoteka na osnovu čega se utvrđuje da li ona treba da nastavi svoj put prema čvoru CCC ili ne. U našem primeru datoteka treba da nastavi svoj put, pa se prebacuje u izlazni red ka čvoru CCC. Od čvora BBB do čvora CCC prenose se jedna po jedna sve datoteke koje čekaju u izlaznom redu čvora BBB.

Tokom puta od čvora do čvora, većina čvorova, putem interaktivnih poruka, šalje obaveštenja o tome da je datoteka prošla kroz taj čvor. Interaktivne poruke se upućuje pošiljaocu datoteke i ispisuje na ekranu njegovog terminala. Na taj način moguće je pratiti kretanje datoteke tokom njene putanje i „prisustvovati" trenutku njenog stizanja na odredište.

Po prijemu na čvoru CCC, na osnovu sadržaja zaglavlja datoteke, konstatuje se da je ona stigla na računar koji je njeno odredište. Takva datoteka se smešta u centralno prijemno područje računara i tamo čeka da bude obrađena (primljena) od strane korisnika kome je upućena.

Ovako opisani način prenosa datoteka nije uobičajena stvar u svetu VAX/VMS mašina. Logika po kojoj radi VAX/VMS u BITNET-u je ona po kojoj radi BITNET, a VAX/VMS/JNET se uklapa kako bi obavio funkcije koje se od njega traže na u ovoj mreži uobičajeni način.

PRIJEM DATOTEKE

Datoteka koja se nalazi u prijemnom području računara ne opterećuje korisnikovu diskvotu. Takve datoteke se obično brišu ukoliko ih niko ne pokupi u roku od petnaest dana. Korisnik preuzima datoteku iz centralnog prijemnog područja pomoću uslužnog programa RECEIVE. Prijem datoteke sa RECEIVE znači onu poslednju kariku u prenosu datoteke: datoteka se smešta u prostor na disku koji kontroliše korisnik koji je datoteku primio. Na nekim mašinama taj prostor se zove katalog, na nekim drugim minidisk. Po prijemu sa RECEIVE datoteka postaje „primljena", nalazi se među svim ostalim datotekama koje korisnik poseduje i može se obrađivati editorom i raznim drugim komandama i uslužnim programima.

STANDARDNI FORMATI

Kao što smo govorili na početku priče o BITNET-u, ova mreža je zasnovana na dosta starim protokolima koji su nasledili neke još starije protokole. Naša priča o RJE stanici iz prošlog nastavka imala je za cilj da vam pokaže kakve su bile komunikacione potrebe onoga vremena. RJE i NJE su bili namenjeni za prenos podataka između glavnog računara i velikog broja čitača kartica (ulazni uređaji), bušača kartica (izlazni uređaji) i štampača (izlazni uređaji).

Na jednu karticu staje 80 znakova u jednom redu. Taj format pošiljuju i čitač i bušač kartica. Format prenošenja podataka putem zapisa čija dužina ne prelazi 80 znakova naziva se *punch format*. Taj format pošlovali su otkako su uvedeni i terminali koji ispisuju (zamislite baš) do 80 znakova u redu. U *punch* formatu moguć je i prenos binarnih podataka. Tada svi zapisi imeju fiksnu dužinu, a svaki osmobitni znak se tumači kao binarni podatak.

Za prenos redova teksta koje treba ispisati na štampaču trebalo je imati mogućnost slanja nešto većih zapisa. Uobičajeno je da na jedan list linijskog štampača staje do 132 znaka u jednom redu. Za štampače je stoga bio predviđen drugi način prenošenja podataka koji je logično dobio ime *print* format. Print formatom je predviđeno prenošenje još jednog dodatnog znaka koji se prenosi na početku svakog zapisa i služi za kontrolu mehanizma za prelazak u sledeći red tokom štampanja. Zajedno sa tim znakom u jednom zapisu datoteke u *print* formatu može biti najviše do 133 znaka.

SPECIJALNI FORMATI

Ono što je u početku izgledalo kao izuzetno dobro rešenje pokazalo se kasnije kao problematično. Kada su se pojavile potrebe prenošenja podataka između bitno drugačijih uređaja, recimo sa diska na disk, standardni formati su se pokazali kao problematični. Kako, recimo, preneti datoteku koja ima zapise duže od više hiljada bajtova? Kako preneti indeks-sekvencijalnu datoteku?

Za prenos komplikovanih zapisa IBM konsti

će povećati broj lokacija koje imaju kvalitetne komunikacione veze i kvalitetan pristup do Univerzitetskog računskog centra, a time i do usluga velikih svetskih globalnih mreža.

U toku je i procedura za nabavku softvera za većinu lokacija koje trenutno ostvaruju pristup do BITNET-a posredstvom mrežnog prolaza, kako bi se i one uključile u BITNET kao punopravni članovi. Time bi se broj pravih BITNET računara u Srbiji uvećao za red veličine.

Pristup do Zavoda za statistiku

Osim pristupa iz prostorija zavoda i iz institucija koje imaju iznajmljene telefonske linije do njega, Nacionalni čvor je dostupan i preko komutiranih telefonskih veza. Na žalost, to nisu veze na kakve su navikli korisnici BBS-ova. Potrebno je da imate dodatnu SDLC karticu, softver za emulaciju terminala IBM 3270 i eksterni modem koji ima mogućnost sinhronog rada. Radi se na tome da se omogući pristup pomoću uobičajenih asinhronih modema, kao i pristup preko JUPAK-a.

Informacije o načinima pristupa i mogućnostima otvaranja radnih naloga na Nacionalnom čvoru možete dobiti na telefonu 011-419 895.

Pristup do Elektrotehničkog fakulteta

Kada je sva oprema ispravna, što je zbog neadekvatne finansijske podrške i problema oko servisiranje sve ređe slučaj, instalacije na Elektrotehničkom fakultetu su Beogradu dostupna je preko 16 telefonskih modemskih linija na kojima vas čekaju V.22bis modemi sa korekcijom grešaka i kompresijom tipa MNP5. Podesite parametre u komunikacionom programu na uobičajene u svetu BBS-ova: 8/N/1, a tip terminala koji se emulira izaberite iz sledećeg spiska: VT100, VT200 ili VT300. Brojevi telefona sa modemima su 327-311 do 327 318 i 339 281 do 339 288.

Do UBBG-a (YUBGEF51), o kome je u ovom tekstu uglavnom reč, stiže se posle nekoliko koraka. UBBG je jedan od računara u lokalnoj mreži Elektrotehničkog fakulteta. Na telefonskim brojevima na dežure UBBG već računar specijalizovan za opsluživanje terminala u lokalnoj mreži (terminalski server). Njemu je potrebno da saopštite ime računara kome interaktivno pristupate.

Procedura ide ovako. Po uspostavljanju veza otkucati dva puta taster RETURN>. Na ovo reaguje terminalski server sa zahtevom ENTER USERNAME>. Otkucajte svoje ime, svoje prezime ili svoj nadimak. U svakom slučaju ništa bezobrazno ni provokativno. Server će ispisati svoju standardnu poruku Local>. Zatim je potrebno da izaberete računar na kome ćete raditi. To se postiže komandom CONNECT UBBG. Sada se javlja UBBG i traži od vas ime korisničkog naloga (Username). Ukoliko ste prethodno obavili sve administrativne formalnosti, tada vi znate šta ovde treba kucati. Posle ovoga UBBG pita za lozinku (Password). Kucaju se podaci dobijeni prilikom otvaranja naloga. Obratite pažnju: za vreme kucanja lozinke nema eha na ekranu. Procedura prijavljivanja završava se kada računar ispiše u prvoj koloni znak $, što je uobičajena oznaka na VAX/VMS u da se od vas očekuje da otkucate komandu.

Rad sa UBBG-om završava se tako što, kada na ekranu u prvoj koloni tekućeg reda imate komandni prompt $, otkucate komandu LOGOUT. UBBG će dati obaveštenje da je završio sesiju sa vama, a vezu ponovo preuzima terminalski server za slučaj da želite da uspostavite novu sesiju sa istim ili nekim drugim računarom iz mreže. Rad se konačno završava tako što terminalskom serveru date komandu LOGOUT (dakle po drugi put LOGOUT). Terminalski server će posle ove komande raskinuti telefonsku vezu sa vama.

Drugi način interaktivnog pristupa do UBBG-a je preko JUPAK-a. UBBG je na broju 1111 0002060.

Dodatne informacije možete dobiti na telefonu Računskog centra Elektrotehničkog fakulteta u Beogradu 011 328-668.

Zahtevi za otvaranje naloga na UBBG u primaju se kod operatera u RC ETF. Mašina i terminalska sala ovog računskog centra dostupni su korisnicima svaki dan, 24 sata dnevno, non-stop.

Disk dump format Ovaj format se zasniva na korišćenju *punch* formata kao nosioca binarnih podataka pri čemu se ne prenose samo podaci nego i opis datoteke. *Disk dump* je format koji omogućava efikasan prenos podataka na bazi punch formata između dve IBM-ove mašine i tu se njegova primena završava.

Rešenje koje je manje zavisno od hardvera razvijeno je nešto kasnije. Poznato je pod imenima **interactive Data Transmision Facility** ili **Netdata**. Format je tipa „puno opisa podataka garnirano sa samim podacima" ali sve u obliku koji se lako može opisati i implementirati i na drugim mašinama. Ovaj način prenosa datoteka postao je ubedljivo najpopularniji na BITNET mreži. Podržavaju ga sve vodeće implementacije NJE na BITNET-u: CMS na VM-u, TSO na MVS-u, JNET i ANJE na VAX/VMS-u i UREP na UNIX-u.

Treći često korišćeni specijalni format je VMSDUMP. Ovaj format prenosa moguće je koristiti samo pri prenosu podataka između dve VAX/VMS mašine na kojima radi JNET softver. Karakteristično mu je da prenosi sve VAX/VMS RMS atribute, pa datoteke prenete na ovaj način predstavljaju potpuno vernu kopiju početne datoteke.

Ako vas čudi otkud sad posebno isticanje činjenice da se prenosi verna kopija (a kakva bi još kopija datoteke mogla biti?), vreme je da se suočite sa tužnom ali istinitom činjenicom da kada se datoteke prenose između računara koji imaju različite organizacije sistema datoteka (File system) nije uvek moguće preneti datoteku tako da ona predstavlja funkcionalno vernu kopiju početne datoteke. Čak i kada su u pitanju najprostije moguće strukture datoteka – sekvencijalne datoteke – sa dobijanjem datoteke koja odražava funkcionalno vernu sliku polazne datoteke može biti problem.

ODREĐIVANJE FORMATA

Već smo rekli da se za izvođenje operacije koju smo nazvali slanje datoteke koristi komanda koja predstavlja engleski ekvivalent naše konstrukcije „slanje datoteke"

$ SEND/FILE naziv datoteke user@node

Međutim, jedine datoteke koje ćete sigurno bez problema poslati kroz mrežu sa ovom bazičnom komandom su sekvencijalne datoteke koje sadrže linije ASCII teksta dužine do 80 znakova. Svi ostali tipovi datoteka doživeće izobličenja tokom prenosa!!

U navedenoj komandi ostavlja se sistemu da izabere format po kome će spakovati datoteku. Sistem bira format po kome će slati podatke između dva standardna formata i to: ukoliko datoteka koju šaljete ima RMS atribut zapisa **carriage return carriage control** datoteka se šalje u PUNCH formatu. U suprotnom slučaju, datoteka se formatira u PRINT format. E sad, ukoliko zapise duže od 80 znakova pošaljete u PUNCH formatu, oni će biti razbijeni u po dva ili više zapisa. Lom nastupa posle 80-tog znaka ili nešto ranije! Ovo, osim u prostim slučajevima, nije lako vratiti na početno stanje.

Kontrolu nad formatom u kome se šalju podaci ostvarujemo specificiranjem formata za slanje. Komanda SEND/FILE se menja u neku od sledećih komandi:

$ SEND/FILE/PUNCH
$ SEND/FILE/PRINT
$ SEND/FILE/NETDATA
$ SEND/FILE/VMSDUMP

TRANSLACIJA KODA

Tu, naravno, nije kraj komplikacijama. Običaj je da se duž BITNET-a prenose EBCDIC datoteke. Radi se o osmobitnom kodu za predstavljanje znakova koji je u poslednjih desetak godina konačno potisnut od nadirućeg ASCII koda, pa se ranije toliko česti problemi oko konverzije dva koda sve ređe sreću.

Kako u takvoj mreži rade velike IBM mašine? Vrlo prosto. Datoteke koje sadrži tekst šalje se nekim od ekvivalenata gore navedenih naredbi. Pošto datoteka i u originalu sadrži tekst u EBCDIC kodu sve je u redu. Kada se šalje izvršna verzija programa ili neka druga binarna datoteka opet nema problema. Datoteka se šalje takva kakva već jeste na velikoj mašini.

Kako to izgleda u slučaju VAX/VMS i UNIX mašina? Pa, recimo da ima problema. Pošto se u NJE mrežama normalno najčešće prenose tekstovi (od čitača kartica do štampača...) komanda SEND/FILE u svim varijetetima koji su do sada spomenuti podešena je tako da vrši konverziju svakog bajta koji se šalje tako što ga tumači kao ASCII znak i konvertuje u odgovarajući EBCDIC znak. To je u redu kada preko mreže prenosite tekst, ali ne i kada šaljete datoteku tipa EXE. Konverzija ASCII > EBCDIC apsolutno sigurno uništava vašu binarnu datoteku bez mogućnosti oporavka.

Da biste poslali binarnu datoteku potrebno je da na mestima gde ste kucali SEND/FILE otkucate SEND/FILE/BINARY, čime uslužnom programu za slanje datoteke dajete informaciju da ne vrši nikakvu translaciju koda. Na primer, za slanje binarne datoteke u NETDATA formatu daćete komandu sledećeg oblika:

$ SEND/FILE/NETDATA/BINARY -
naziv_datoteke -
user@node

USLUŽNI PROGRAM RECEIVE

Za prijem datoteke koristi se uslužni program RECEIVE. Program se pokreće istoimenom komandom. Odmah po pokretanju RECEIVE ispisuje spisak datoteka koje su iz mreže stigle na vaše ime, a zatim se oglašava promptom **RECEIVE>** čime vas podseća da treba da mu date neku od komandi koje ovaj uslužni program prihvata. Potpuni spisak komandi koje prihvata RECEIVE dobija se ako otkucate komandu HELP. Iz uslužnog programa RECEIVE izlazi se davanjem komande EXIT.

Najčešće korišćene komande uslužnog programa RECEIVE su COPY, RECEIVE, TYPE, DELETE i DIRECTORY.

$ RECEIVE

...

RECEIVE> DIRECTORY

Komanda DIRECTORY daje spisak svih datoteka u centralnom prijemnom području, a koje čekaju da ih korisnik koji daje komandu primi. Pored naziva datoteke (*Source file*) ispisuju se i sledeće informacije kao na slici 2.

Class je klasa u koju spada datoteka. Datoteke se klasifikuju na dva načina: po standardnom formatu u kome su poslate i po klasi znakova koja je korišćena. Za prenos datoteka koriste se dva osnovna standardna formata: PUNCH i PRINT. Na mestu „klasa znakova" nalazi se jedan alfanumerički znak. Ovaj znak se može po volji definisati prilikom slanja datoteke. Ipak, klase znakova koje ćete najčešće sresti u RECEIVE području su: A – što označava i ASCII i NETDATA datoteke i N – što označava binarne datoteke. **Node** je ime čvora sa koga je došla datoteka. **User** je korisnički nalog koji je korišćen prilikom slanja datoteke. **Date** i **Time** označavaju datum i vreme kada je datoteka poslata. **Records** označava broj zapisa. U slučaju datoteke koja sadrži tekst, to je broj redova teksta unutar datoteke. Ovaj podatak se može tretirati kao pokazatelj na osnovu koga se može nešto zaključiti o veličini datoteke, ali se, na žalost, ne može zaključiti previše jer svi zapisi ne moraju biti (i obično nisu) iste veličine.

RECEIVE> TYPE imedatoteke.tip;verzija

Komanda TYPE je jedina komanda pomoću koje je moguće da se ispiše sadržaj primljene datoteke na ekranu, u vreme dok je datoteka još uvek u centralnom prijemnom području. Obratite pažnju! Moguće je da „zaglavite" terminal!

RECEIVE> DELETE imedatoteke.tip;verzija

Komanda DELETE briše datoteku iz centralnog prijemnog područja.

RECEIVE> COPY imedatoteke.tip;verzija

Komanda COPY kopira sadržaj datoteke iz centralnog prijemnog područja u korisnikov katalog. Originalna datoteka i dalje ostaje i u prijemnom području.

RECEIVE>RECEIVE imedatoteke.tip;verzija

Komanda RECEIVE kopira sadržaj datoteka iz centralnog prijemnog područja u korisnikov katalog i, posle toga, briše datoteku iz centralnog prijemnog područja.

Obe upravo pomenute komande za prijem datoteka (i COPY i RECEIVE) znaju da prepoznaju specijalne formate (NETDATA, VMSDUMP). Raspakivanje specijalnih struktura vrši se automatski i tu korisnik ne mora da vodi mnogo računa.

Međutim, u oba slučaja podrazumeva se i to da je datoteka koja je stigla prepuna EBCDIC teksta i da je treba konvertovati u ASCII. To apsolutno treba sprečiti ukoliko datoteka ne sadrži EBCDIC tekst. Dakle, ako je primljena izvršiva datoteka, nekakava arhiva ili bilo koja druga vrsta binarnog dokumenta, konverzija se ne sme izvršiti. To se saopštava tako što uz komande za prijem datoteke treba navesti modifikator komande /BINARY. Dakle:

RECEIVE> COPY/BINARY imedat.tip;ver ili
RECEIVE> RECEIVE/BINARY imedat.tip;ver

Jedan od načina da pre prijema utvrdite da li je datoteka stigla u EBCDIC-u je da date komandu

RECEIVE> TYPE ime.tip;verzija

Ovako zadata komanda TYPE vrši konverziju EBCDIC -> ASCII pre ispisivanja na ekran. Ukoliko je tekst čitak, pri prijemu treba vršiti konverziju. Ukoliko je tekst nečitak možete probati sa sledećom komandom

RECEIVE> TYPE/BINARY ime.tip;verzija

U ovom slučaju ne vrši se konverzija pre ispisivanja. Datoteku čitljivu na ovaj način treba primiti sa RECEIVE/BYNARY ili sa COPY/BYNARY. Međutim, datoteka ne mora biti čitljiva ni na jedan od navedena dva načina. Tada se verovatno radi o nekoj arhivi ili programu u izvršnom obliku, pa je velika verovatnoća da ga treba primiti kao binarnu datoteku. Ponovo obratite pažnju: Komanda TYPE korišćena na podacima čiju prirodu ne znamo može zaglaviti terminal.

Da li je ovo kraj komplikacije zvane prenos datoteka preko BITNET-a? Na žalost ne. To su samo elementi koje treba znati da bi se razumele komplikacije do kojih dolazi pri prijemu datoteka koje šalju serveri koji komuniciraju sa drugima mrežama. Komande koje smo naveli su samo deo alata koji svaki korisnik mora da nauči da bi što naprednije koristio BITNET.

PREKO PREČE NAOKOLO BLIŽE

Čovekova sklonost da egzaktno formuliše i pred svoj um postavlja nove probleme može se uporediti samo sa njegovom strašću da ih rešava. Problem najkraćeg puta ne predstavlja zaštitni znak matematike poput kvadrature kruga ili trisekcije ugla, niti može da posluži kao ključni dokaz (be)smislenosti neke filozofske koncepcije poput Zenonove aporije o Ahilu i kornjači, ali zato na jedan upečatljiv način pokazuje kako u osnovi rešenja mnogih raznorodnih matematičkih i logičkih problema leže jedinstveni zakoni prirode.

Saša Marković

U ovom tekstu će biti prikazana *metoda simuliranog očvršćavanja*, rafinirana kompjuterska tehnika za rešavanje široke klase optimalnih problema. Ova metoda je efektivno rešila čuveni *problem najkraćeg puta* (*problem trgovačkog putnika*) pronalazeći najkraću zatvorenu putanju između *N* slučajno razmeštenih tačaka. U praksi se danas intenzivno koristi za projektovanje kompleksnih integrisanih kola gde raspored elektronskih komponenti treba da obezbedi minimalnu dužinu spojnih vodova uz minimalan broj ukrštanja. Oba navedena zadatka iz domena kombinatorne optimizacije mogu se rešiti istovetnim postupkom.

FORMULACIJA PROBLEMA

Problem najkraćeg puta samo na prvi pogled predstavlja lak zalogaj. Neka su u ravni zadate koordinate položaja *(x, y)* *N* slučajno razmeštenih gradova numerisanih brojevima 1, ... *N*. Trgovački putnik treba da tačno jednom prođe kroz svaki grad i na kraju se vrati u polaznu stanicu, uz uslov da pređeni put bude minimalan. Očigledno je da svako potencijalno rešenje, odnosno redosled obilaženja gradova, predstavlja jednu permutaciju od *N* elemenata ili, apstraktno rečeno, jednu tačku u *N*-dimenzionalnom prostoru. Skup svih mogućih redosleda obilaženja predstavlja konfiguracioni prostor našeg problema. Ovaj prostor je diskretan jer se ne radi o prostoru „razapetom" nad *N* neprekidnih promenljivih parametara. Bolje rečeno, u našem problemu se radi o konfiguracionom prostoru koji se sastoji iz veoma velikog broja diskretno raspoređenih tačaka. Za svaku tačku ovakvog prostora može se sračunati vrednost neke adekvatno definisane funkcije. Ako iz čitavog konfiguracionog prostora treba izdvojiti onu tačku za koju funkcija ima apsolutni minimum (ili maksimum) kažemo da se radi o objektnoj funkciji. Jasno je da u našem problemu broj tačaka konfiguracionog problema može biti ogroman (reda *N*!) a da je objektna funkcija zapravo pređeni put koji se može izračunati za svaki redosled obilaženja.

Sirova upotreba računarske snage ovde ne pomaže. Čak i u slučaju svega pedeset gradova broj različitih redosleda obilaženja je tako veliki da klasične metode nalaženja minimuma ne vode rešenju. Može se pokazati da vreme potrebno za nalaženje egzaktnog rešenja raste sa faktorom $e^{const \cdot N}$ pri čemu sa porastom broja *N* ono vrlo brzo premaši računarski potencijal. Takođe, problem najkraćeg puta spada u probleme sa mnogo lokalnih minimuma. Naime, postoji mnogo redosleda obilaženja (konfiguracija) sa karakterom lokalnog ali ne i apsolutnog minimuma. U realnim problemima, međutim, najčešće je dovoljno između tih ekstremuma izabrati onu konfiguraciju koja se dalje ne može značajno popraviti. *Postupak simuliranog očvršćavanja* upravo ima za cilj da pomoću niza iterativnih rekonfiguracija putanja pronađe „dovoljno" optimalno rešenje, dakle redosled obilaženja koji se više ne može unaprediti bez enormnog utroška vremena.

FIZIČKA ANALOGIJA

U srcu ove kompjuterske tehnike, verovali ili ne, leži analogija sa termodinamikom, odnosno sa procesima smrzavanja i kristalizacije. Na dovoljno visokim temperaturama molekuli tečnosti se slobodno kreću. Ako temperatura opada smanjuje se i pokretljivost molekula pri čemu na temperaturi očvršćavanja dolazi do njihovog grupisanja u uređene prostorne modele kristalnih rešetki. Iako idealna kristalna forma sadrži u sebi na bilione molekula, ona se odlikuje strogom geometrijskom pravilnošću na rastojanjima koja su nemerljivo veća od dimenzija pojedinačnog molekula. Kristalna struktura predstavlja stanje minimalne potencijalne energije čitavog sistema. Zadivljuje „sposobnost" prirode da između nebrojeno mnogo različitih konfiguracija pronađe baš onu sa minimalnom energijom. U suštini ovog fenomena leži proces postepenog, laganog hlađenja i očvršćavanja koji dozvoljava sinhronizovano razmeštanje molekula u pravilne geometrijske forme. Ako se tečnost naglo ohladi, molekuli ne stižu da se razmeste u čvorove kristalne rešetke tako da nastaju polikristalne ili amorfne čvrste forme povišene energije.

I ovde stižemo do ključnog pitanja: na koji način priroda vrši optimizaciju rasporeda molekula nepogrešivo pronalazeći stanje minimalne energije sistema? Odgovor leži u statističkoj raspodeli energija pojedinih molekula. Verovatnoća da molekul poseduje energiju *E* na temperaturi *T* proporcionalna je faktoru $e^{-E/kT}$ što znači da i u uslovima niskih temperatura određen broj molekula ima visoku energiju odnosno veliku pokretljivost. Zahvaljujući postojanju ovakvih molekula sistem može slobodno da „šeta" između različitih energetskih stanja i da sa određenom verovatnoćom napusti stanje lokalnog energetskog minimuma i potraži globalno povoljnija rešenja. Međutim, sa opadanjem temperature ova verovatnoća se smanjuje.

PRINCIPSKE OSNOVE METODE

Godine 1953. **Metropolis** i saradnici su ugradili ove principe u razne numeričke proračune. Iako je analogija u početku korišćena samo za simuliranje termodinamičkih procesa, vrlo brzo su formulisani opšti elementi ovakvog modeliranja:

- **Adekvatan opis konfiguracija sistema**: u problemu najkraćeg puta to je permutacija brojeva 1, ... *N* posmatrana kao redosled obilaženja gradova.
- **Generator slučajnih promena u konfiguraciji**: u našem problemu treba definisati način na koji se iz jednog redosleda obilaženja dobija drugi. Ove slučajne rekonfiguracije analogne su termičkom kretanju molekula. Na slici 1 prikazana su dva ravnopravno korišćena postupka: (1) deo putanje se ukloni

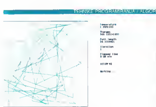


Slučajne promene redosleda obilaženja (prema Linu)

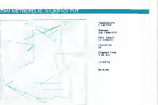
Listing 1

```
TRAVEL.C   THE TRAVELLING SALESMAN PROBLEM
```

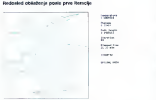
Problem trgovačkog putnika rešen algoritmom Metropolis



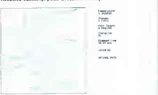
Redosled obilaženja posle prve iteracije



Redosled obilaženja posle dvadesete iteracije



(Skoro) optimalan redosled obilaženja



Optimalni redosled obilazenja sa dopunskim ograničenjima

a zatim zameni istim gradovima obilaženim u suprotnom smeru i (2) deo putanje se ukloni a zatim umetne između dva slučajno izabrana grada na nekom drugom delu putanje. Sa slike se vidi da optimalna putanja ne može da ima ukrštenih segmenata jer je AC+BD>AB+CD

- **Objektna funkcija kao analogija energiji sistema *E*** u našem primeru to je pređeni put. Neka je pređeni put za posmatrani redosled obilazenja E_1, a za neki variant E_2. Ako je $E_2<E_1$ uvek ćemo trenutnu konfiguraciju zameniti ispitivanom. Ako je $E_2<E_1$, zamena će biti izvršena sa verovatnoćom $e^{-(E_2-E_1)/kT}$. *To znaži da ćemo sa određenom verovatnoćom namerno uzimati nepovoljniji redosled obilaženja sa tendencijom da sa opadanjem temperature takva pogoršanja konfiguracije potpuno eliminišemo.* Ovo je suština *Metropolis*-algoritma. Pod uspešnom rekonfiguracijom podrazumevaćemo svaku zamenu postojećeg novim redosledom obilaženja
- **Procedura „smrzavanja"** ovde je eksperiment neophodan kako bi se utvrdio postupak koji vodi dobijanju optimalnog rešenja. Pre početka optimizacije izvršićemo nekoliko nasumičnih rekonfiguracija prateći promenu pređenog puta delta E. Maksimalnu promenu pređenog puta ćemo uzeti za početnu vrednost parametra („temperature") *T*. Nakon svake iteracije temperaturu ćemo smanjiti za 10% i tako smanjiti verovatnoću sa kojom dozvo-ljavamo namerno kvarenje konfiguracije. Za svaku iteraciju odnosno za svaku vrednost temperature *T* ćemo izvršiti 100*N* rekonfiguracija pri čemu ćemo pratiti broj uspešnih rekonfiguracija. Ukoliko on premaši 10*N* temperaturu automatski smanjujemo i prelazimo na narednu iteraciju. Čitav postupak ćemo obustaviti kada se rešenje na određenoj temperaturi smrzne, odnosno kada broj uspešnih rekonfiguracija padne na nulu

RASPLITANJE GORDIJEVOG ČVORA

Priloženi program napisan za **TurboC** ubedljivo demonstrira kvalitet ovog algoritma. Na slikama 2 3 4 prikazan je postupak „kristalizacije" optimalnog rešenja za 50 gradova slučajno razmeštanih unutar jediničnog kvadrata. Obratite pažnju na postupno opadanje temperature i na dva brojača rekonfiguracija: u dvadesetoj iteraciji je 244 puta prihvaćena rekonfiguracija koja je vodila smanjenju pređenog puta a 256 puta rekonfiguracija koja je pređeni put povećavala. Ako vrednost preprocesorske konstante **CONSULT** promenite u nulu i tako potpuno eliminišete ovu drugu mogućnost, videćete da program ne može da pronađe optimalno rešenje već se beznadežno zaglavi u nekom od lokalnih minimuma. Za one koji vole da eksperimentišu reći ću da **NCITY** predstavlja broj gradova unutar jediničnog kvadrata a **STARTSEED** „seme" za generator slučajnih brojeva. Rekonfiguraciju redosleda obilaženja vrše potprogrami *reverse* (inverzija redosleda) i *transport* (isecanje i povezivanje). Objektnu funkciju definiše potprogram *objectfunction* sa kojom se slobodno poigrajte. Tako je na slici 5 prikazana optimalna putanja za isti raspored gradova pri čemu se za svaki prelazak preko dijagonale kvadrata pređeni put uvećava za 2. Uporedite ovo rešenje sa onim sa slike 4 gde se ovakvi prelasci ne sankcionišu. Potprogram *Testimate* procenjuje početnu vrednost parametra *T*, a potprogram *ran* (**Knuth**) predstavlja izvrstan portabilan generator slučajnih brojeva. Potprogrami *setallpositions* i *setposition* razmeštaju gradove unutar jediničnog kvadrata. U srcu celog programa čuči potprogram *optimize* koji prema opisanoj proceduri snizava temperaturu u računaru i na kraju isporučuje zamrznuto rešenje. Ostali potprogrami uglavnom služe za elementarnu grafičku prezentaciju međurešenja i osnovnih podataka iz proračuna. Na kraju kratka *main*-rutina sve to objedinjuje u jednu celinu koja rekao bih izgleda šašavo ali deluje!

JEDNA MALA SUGESTIJA

Za sada nema pouzdanog dokaza matematičke superiornosti prikazanog postupka u proizvoljnom praktičnom slučaju. Međutim neke vrline ovog postupka su očigledne. Pre svega algoritam je vrlo teško zavarati postojanjem lokalnih minimuma što je generalni problem kod mnogo poznatijih optimizacionih tehnika. Postupak nalaženja optimalnog rešenja slobodno osciluje između nekoliko lokalnih ekstremuma unutar dijapazona širine *T* pri čemu se broj posećivanih konfiguracija značajno smanjuje sa opadanjem temperature. Osim toga postupak optimizacije teče na prirodan način: drastična redukcija pređenog puta odvija se na visokim temperaturama da bi se pri kraju vršila samo suptilna poboljšanja optimalne putanje. U problemu na slici 5 program vrlo rano donosi odluku da kritičnu dijagonalu prađa samo dva puta a zatim fino optimizuje delove putanje unutar gornjeg i donjeg polukvadrata

Recimo na kraju da smo ovde tek načeli jednu široku temu: kako „opšte i jednostavne" zakone prirode implementirati na računaru i upotrebiti ih u rešavanju analognih praktičnih problema. Ova oblast još uvek čeka svog Ajnštajna. Zato ugasite barem za trenutak sve te vaše 486-ice, kolor monitore i laserske štampače, otvorite prozor i pogledajte negde preko reke. Priroda je već davno rešila mnoge probleme, verovatno je i vas negde zakačila. Treba samo uporno da gledate

UNUTRAŠNJI MOTOR

Ovo su teška vremena za Prave programere. Nikoga više ne zanima koliko dobro možete da uradite ručnu optimizaciju koda dobijenog iz C kompajlera, ili koliko nanosekundi možete da uštedite pišući samomodifikujući kod unutar tesne petlje. Danas poslodavci umesto simulacija subatomskih procesa u realnom vremenu traže baze podataka sa lepim sličicama, a klinci sa *Clipper*-om ih štancuju kao na pokretnoj traci. Ovaj tekst je tu da pokaže da Pravi programeri ipak to rade bolje.

Dejan Jelović

Još uvek se sećam svog prvog susreta sa *Clipper*-om: automatska deklaracija promenljivih, nestrukturirani programi i idiotska sintaksa nasleđena iz dBASE-a! Bio je to zapravo bejzik za rad sa bazama podataka. Instinktivno sam rekao sebi da „to može i bolje". A onda sam seo da to i napravim.

OSVRT NA STARO

Iako je kritika *Clipper*-a postala nacionalni sport C programera, on još uvek predstavlja jedan od bržih načina da se napravi baza podataka. Njegove glavne mane nastale su iz, da tako kažem, istorijskih razloga. Pošto je nastao kao dBASE prevodilac, morao je da za osnovu uzme dBASE sintaksu. Onda su autori, usavršavajući program, dodali mnoga poboljšanja koja su, zbog kompatibilnosti sa prethodnim verzijama, ispala obične zakrpe. *Clipper* je osuđen: ako ne odustane od kompatibilnosti sa starijim verzijama, nikada se neće razviti u moderan, efikasan jezik.

Ipak, i pored svih svojih mana, *Clipper* ostaje dominantna alatka za pravljenje baza podataka. Njegovu prednost čine, pre svega, biblioteke koje se uz njega dobijaju. *Clipper* programeri ne moraju da se bave kodom za crtanje maski za unos, komunikaciju sa bazom ili indeksiranjem, već mogu da se skoncentrišu na konkretan zadatak pisanja programa. Tu negde leži i rešenje: ukoliko bismo nekom „pravom" programskom jeziku dodali biblioteke funkcionalno odgovarajuće onima koje ima *Clipper*, dobili bismo najbolje iz oba sveta.

U izbor jezika na kojima valja napraviti ove biblioteke ulaze Turbo paskal, C i C++ kao popularni jezici današnjice. Izbor je pao na ovaj poslednji, uglavnom kao rezultat mojih afiniteta. Pri tom, mnogi C programeri koriste ovako razvijene C++ klase u svojim programima bez ikakvog znanja o objektnom programiranju. Jednostavno, klasa predstavlja mnogo lepši način da se zapakuje biblioteka. Šta je unutra, to nikoga ne zanima.

IMAJTE NA UMU

Prvo želim da navedam nekoliko „zakona" kojih bi trebalo da se pridržavate pri pravljenju baza podataka. Ako me ne budete poslušali, verovatno nećete otići u zatvor. Ali će vam duša goreti u paklu, imaćete lošu karmu i, što je najgore – vaši programi neće raditi. Dakle:

- **Nema malih baza podataka.** Jednostavno, svaka baza podataka ima tendenciju da raste. Klijent će vas pozvati i zamoliti da dodate još par opcija, zahtevaće da nešto promenite, tražiće nove tipove izveštaja itd. Zato svaku bazu podataka treba praviti tako da se ostavi prostor za lake izmene i dopune.
- **Nikada podatke ne držite u memoriji.** Jednostavno, memorija je previše mala za čuvanje baze ili indeksa (vidi prethodnu tačku). Baš je skoro bio aktuelan problem zbog nepridržavanja ovog principa. Jedan je moj prijatelj uzeo da, radi brzine, drži indeks baze podataka u memoriji, koja je za tu svrhu delovala kao sasvim dovoljna. Međutim, njegov revnosni klijent je taj indeks popunio za nekih godinu dana i aplikacija je pala. Pouka: ostaje, svi podaci se drže na disku, u memoriji se ne čuva ništa.
- **Uvek koristite long i double tipove podataka.** Kako baze često mogu da narastu preko nekih 32 hiljade podataka, što je opseg običnih **int**-ova, uvek se za rad sa bazama koriste **long int**-ovi. Drugi zahtev, korišćenje *double float* brojeva, odnosi se na polja unutar baza. Da se razumemo, baze podataka najčešće služe za baratanje novcem, a obični *float* brojevi jednostavno ne mogu da opišu neku malo veću svoticu. (Nadajmo se samo da sa današnjom inflacijom nećemo morati da pređemo na neki još veći format.)
- **Razradite mehanizam za reagovanje na greške.** Često se dešava da se u programu zanemare rezultati I/O operacija, što može da dovede do gubljenja podataka. Da li vam se žmarci penju uz kičmu kada pomislite da vaš klijent može da izgubi sto hiljada maraka zbog greškice u vašem programu?

ZAPIS I ČITANJE SLOGA

Osnova svake baze podataka je deo koji upisuje i čita slogove sa diska. Iako mi o bazama podataka razmišljamo uglavnom u terminima polja i njihovog značenja, za ovaj deo uopšte nije bitno kako slog izgleda. Sve sa čime operiše je dužina sloga, i ima samo dve osnovne operacije: prepisivanje sloga iz memorije na određeno mesto na disku i prebacivanje određenog sloga sa diska u računar. Kao (funkcionalan) primer ovog dela pogledajte klasu **database**. Klasa sadrži sledeće članove:

open (char *name, int recsize)

otvara bazu podataka pod imenom **name** i sa dužinom sloga **recsize**. Na primer, ukoliko imamo slog deklarisan sa

```
struct radnik {
        int radnibroj;
        char ime [30];
        char prezime [30];
        double plata;
        int odeljanja;
```

bazu pod imenom RADNICI.DBF bismo otvorili naredbom

open („RADNICI.DBF", sizeof (radnik));

position (long pos)

postavlja pokazivač tekućeg sloga (PTS) na slog pod brojem **pos**. PTS prestavlja redni broj sloga koji će sledeći biti učitan ili upisan.

read (slog *sl)

upisuje tekući slog u **sl** i pomera PTS na sledeći. Ukoliko bismo imali slog r tipa **rednik** učitali bismo tekući slog iz baze naredbom **read (&r)**

write (slog *sl)

upisuje slog **sl** u bazu na mesto na koje pokazuje PTS i pomera PTS na sledeći slog

read (long pos, slog *sl)

učitava slog pod rednim brojem **pos**

write (long pos, slog *sl)

upisuje slog **sl** na mesto pod rednim brojem **pos**

del (long pos)

briše slog pod rednim brojem **pos**. Za svaki slog se na disku odvaja prostor dovoljan da se on upiše plus jedan bajt. Funkcija **write** normalno taj bajt postavlja na 1, a njegovo postavljanje na 0 označava da je slog obrisan

add (slog *sl)

dodaje još jedan slog u bazu. Ova funkcija vraća poziciju na kojoj je slog upisan, a u slučaju da je došlo do greške vraća -1

close ()

funkcija zatvara datoteku sa bazom podataka

currec

promenljiva koja sadrži pokazivač na sledeći slog za upis/čitanje

maxrec

ukupan broj slogova u bazi

U datoteci DATABASE.HPP se nalazi još i funkcija **build base**. Njena svrha je da kreira praznu bazu podataka na disku. Jedan konstruktor poziva funkciju **open**, a drugi ne radi ništa. Destruktor zatvara bazu u slučaju da je otvorena. Pri tom sve navedene funkcije vraćaju vrednost tipa **int**, koja je 1 u slučaju da je operacija uspela, 0 u slučaju da nije, a kod greške se smešta u promenljivu **errno**. Dve definisane greške su **D_IO**, koja pokazuje da je došlo do greške pri radu sa ulaznom datotekom, dok **D_EOF** vraća funkcija **read** ako u bazi nema više slogova.

Uobičajen pristup dodavanju novih slogova je da se oni stavljaju na kraj datoteke, a da se „rupe" nastale brisanjem popunjavaju periodičnim pakovanjem baze. Ja sam se odlučio za nešto drugačiji pristup. Ukoliko ima obrisanih slogova, novi slogovi se upisuju preko njih. Tek ukoliko ih nema, novi se slogovi dodaju na kraj datoteke. Brzina ove tehnike se pokazala zadovoljavajućom, a na ovaj način nema potrebe za pakovanjem. Ja u radu koristim navedeno rešenje plus mali keš koji pamti gde se nalaze obrisani članovi. Naravno, ukoliko vam se ovo ne bude sviđalo, funkcija **add** je virtuelna i možete je slobodno menjati.

Listing 1 DATABASE.HPP

```
#ifndef __DATABASE
#define __DATABASE

// Greske
enum dberror {
        D_EOF,
        D_IO
}

void build_base (char *name);

class database {
        protected:
        int recsize;        // Duzina sloga

        public:
        FILE *fp;
        long maxrec, currec;        // Koliko ima slogova u fajlu i trenutni

        database (char *name, int recs);
        ~database (void);
        int open (char *name, int recs);
        virtual int write (void *slog);
        virtual int read (void *slog);
        virtual int position (long);
        virtual int write (long pos, void *slog);
        virtual int read (long pos, void *slog);
        virtual long add (void *slog);
        virtual int del (long);
        int close (void);
        ~database (void);
}

#endif
```

Heder datoteka za *DATABASE.CPP*

Listing 2 DATABASE.CPP

```
#include <stdio.h>
#include <string.h>
#include "database.hpp"

void build_base (char *name)

// Pravi praznu bazu na disku

{
        fclose (fopen (name, "w"));
}

/*************************************************************************

        Implementacija elementarne baze podataka klase

*************************************************************************/

database::database (char *name, int recs)

// Konstruktor klase database

{
        open (name, recs);
}

database::~database (void)

// Destruktor klase database

{
        fp = NULL;
}

int database::open (char *name, int recs)

// Otvori bazu za rad

{
        fp = fopen (name, "r+b");
        if (fp) {
                fseek (fp, 0, SEEK_END);
                currec = maxrec = ftell (fp) / (recsize = recs) + 1;
        }
        else {
                errno = D_IO;
        }
        return fp != NULL;
}

int database::write (void *slog)

// Upisi slog na trenutnu poziciju i pomeri se na sledeci

{
        char c = 1;
        int ret = fwrite (&c, sizeof (c), 1, fp);
        if (ret) {
                if (!(ret = fwrite (slog, recsize, 1, fp)))
                        errno = D_IO;
        }
        else
                errno = D_IO;
        if (++currec > maxrec)
                maxrec++;
        return ret;
}

int database::read (void *slog)

// Ucitaj slog sa tekuce pozicije i predji na sledeci

{
        char c;
```

```
int database::read (void *slog)

// Ucitaj slog sa tekuce pozicije i predji na sledeci

{
        char c;
        int r;

        do {
                #pragma warn -pia
                if (ret = fread (&c, sizeof (c), 1, fp)) {
                #pragma warn .pia
                        if (!(ret = fread (slog, recsize, 1, fp)))
                                errno = D_IO;
                }
                else
                        errno = D_IO;
        } while (!c && ret);
        if (ret)
                if (currec++ == maxrec)
                        errno = D_EOF;
        return ret && c;
}

int database::position (long pos)

// Namesti aktuelni slog na pos

{
        int r = !fseek (fp, pos * (recsize + 1), SEEK_SET);
        if (r)
                currec = pos;
        else
                errno = D_IO;
        return r;
}

int database::write (long pos, void *slog)

// Upisi slog na mesto pos

{
        if (position (pos))
                return write (slog);
        return 0;
}

int database::read (long pos, void *slog)

// Ucitaj slog sa mesta pos

{
        if (position (pos))
                return read (slog);
        return 0;
}

long database::add (void *slog)

// Dodaj slog u bazu, obrati paznju da se popune mesta obrisanih slogova
// Vraca mesto upisanog sloga u bazi

{
        char c;
        for (long i = 0; i < maxrec; i++) {
                if (!position (i))
                        return -1;
                fread (&c, 1, 1, fp);
                if (!c) {
                        if (write (i, slog))
                                return i;
                        else
                                return -1;
                }
        }
        if (write (maxrec, slog))
                return maxrec - 1;
        else
                return -1;
}

int database::del (long pos)

// Brisi slog pos

{
        char c = 0;
        if (position (pos))
                if (fwrite (&c, sizeof (char), 1, fp)) {
                        position (pos);
                        currec++;
                        return 1;
                }
                else
                        errno = D_IO;
        return 0;
}

int database::close (void)

// Zatvara bazu podataka
{
        if (fp) {
                fclose (fp);
                fp = NULL;
                return 1;
        }
        return 0;
}

database::~database (void)

// Destruktor klase database

        close ();
}
```

Osnovne rutine za manipulisanje slogom

KAKO DALJE?

Do sada smo ostvarili deo za baratanje slogovima. Ovaj deo, zajedno sa rutinama za rad sa indeksima, predstavlja ono što bi Amerikanci nazvali **engine**, to jest neku vrstu unutrašnjeg motora baze podataka. Rutine su dovoljno dobre za profesionalan rad i imaju samo jednu manu: ne rade pod mrežom. Funkcije koje bi radile pod mrežom predstavljaju preveliki zalogaj za jedan novinski članak. Ukoliko se potreba za njim javi, na tržištu postoji masa biblioteka koje rade ovaj posao, a vi ih zahvaljujući prednostima objektnog programiranja, možete bezbolno dodati vašim aplikacijama, kod obe klase su funkcije za komunikaciju sa diskom virtuelne i prema tome zamenljive.

Da li ste ovim člankom razočarani? Obećao sam vam lako pisanje baza podataka, a nisam vam dao ni jednu rutinu koja bi crtala one lepe ekrane koje ste viđali kod vaših drugova sa *Clipper*-om. Doći će i to na red - nakon što se u sledećem broju pozabavimo indeksiranjem baze i binarnim stablima. Ovo je bio samo uvod.

Kemal Selećimin

U ovom nastavku ćemo se malo zadržati na delu *Clipper*-a koji se veoma često koristi – na **GET** sistemu za interaktivni unos podataka.

GET sistem

GET sistem je skup naredbi i funkcija koje omogućavaju interaktivno učitavanje vrednosti raznih tipova promenljivih. Ovaj sistem je u *Clipper*-u 5.01 realizovan preko objekata na osnovu primitivnih funkcija tipa „ispiši vrednost promenljive", „pomeri kursor za jednu poziciju". **GET** sistem je realizovan kao obična *Clipper* procedura. Ako vam zatreba, možete promeniti datoteku **GETSYS.PRG** i na taj način napraviti sopstveni **GET** sistem. Međutim, to se vrlo retko radi jer i ovakav sistem uglavnom zadovoljava potrebe.

Najjednostavnija maska za unos se pravi jednostavnim uzastopnim pozivima **GET** naredbi:

```
@ 1,1 GET x
@ 2,1 GET y
READ
```

Kada se „aktivira" naredba **READ**, korisnik može interaktivno da menja vrednosti promenljivih **X** i **Y**. Treba znati da se prilikom unosa ne izvršavaju **GET** naredbe, kako se prilično često misli. **GET** naredbe služe samo za specificiranje promenljivih čije vrednosti treba unositi – za vreme unosa izvršava se isključivo naredba **READ**. Zato sledeća konstrukcija ne radi:

```
@ 1,1 GET x
Beep ()
@ 2,1 GET y
READ
```

Znači, ako očekujete da svaki prelaz sa prve na drugu poziciju za učitavanje (ili obrnuto) aktivira proceduru **Beep**, prevarićete se. Ova procedura se izvršava samo jednom i to prilikom specifikacija promenljivih **X** i **Y**. Stvar postaje nešto jasnija ako se pogleda rezultat rada pretprocesora. Dakle, prevodilac na osnovu prethodnog dela programa „vidi" ovakav tekst:

```
SetPos( 1, 1 )
AAdd( GetList, _GET_( x, "x", ... ) )
Beep ()
SetPos( 2, 1 )
AAdd( GetList, _GET_( y, "y", ... ) )
ReadModal(GetList) ; GetList := {}
```

Vidi se da svaka **GET** naredba pozicionira kursor i izvršava proceduru **aAdd** koja služi za dodavanje elementa na kraj niza. Procedura (funkcija, u stvari) **aAdd** u niz **GetList** dodaje rezultat funkcije _**GET**_ – **objekat** koji sadrži potrebne informacije o unosu promenljive: naziv promenljive, poziciju kursora itd. Naredba **READ** se pretprocesira u poziv funkcije **ReadModal** koja, u stvari, i realizuje interaktivan zbor, a posle toga se niz objekata poništava. U datoteci **GETSYS.PRG**, koja se dobija uz *Clipper*, **nalazi se tekst procedure ReadModal**.

Izostavljanjem naredbe **GetList** := {}, postiže se da niz objekata ostane sačuvan pa će sledeći poziv funkcije **ReadModal** zvršiti isti unos podataka. Isti efekat se postiže naredbom **READ SAVE**. Jedna primena ovoga može biti provera zavisnosti između promenljivih koje se unose bez naknadnog specificiranja **GET** naredbi. Na primer:

```
@ 1,0 GET x
@ 2,0 GET y
WHILE !CorrectInput ()
    ReadModal (GetList)
END DO
```

Sledeći interesantan slučaj je kad se maska za unos „proširuje" novim elementima. Ovo se može koristiti kada se u zavisnosti od nekog polja može javiti i potreba za unosom dodatnih podataka. Na primer:

```
@ 1,0 GET Ime
@ 2,0 GET Zanimanje
ReadModal (GetList)
IF Zanimanje <> „nezaposlen"
    @ 3,0 GET Firma
    ReadModal (GetList)
END IF
```

U slučaju da se unese zanimanje različito od „nezaposlen", maska za unos se proširuje elementom za unos naziva firme i ponovo se vrši unos podataka. Naravno, prethodno unesene vrednosti ostaju sačuvane. Primetite da je sada moguće menjati i prethodno unete podatke (Ime i Zanimanje).

Unos diskretnih promenljivih

Ponekad je promenljiva koju treba uneti diskretnog tipa: može uzimati samo konačan skup vrednosti. Na primer, kao vrednost promenljive **Valuta** može se uneti string „DIN", „DEM" ili „CHF". Proveru da li je korisnik uneo odgovarajuću vrednost možemo izvršiti na više načina, najjednostavnije je pomoću opcija **VALID** i **WHEN**. Ukoliko vrednosti za izbor ima manje, zgodno je omogućiti korisniku da ih izabere iz menija, a ne da ih ukucava. To znači da treba omogućiti da se kod unosa određene promenljive otvori meni. Najjednostavnije je da se kod odgovarajuće **GET** naredbe postavi **VALID** uslov koji otvara meni u slučaju da ukucana vrednost nije korektna: ako korisnik u prethodnom slučaju otkuca „DIN", nastavlja se unos sledeće vrednosti, a ako unese „ABC", otvoriće se meni sa opcijama „DIN", „DEM" i „CHG". Ako su nazivi duži, nema mnogo smisla omogućiti korisniku da bira hoće li otkucati vrednost ili će je izabrati iz menija. Mnogo je bolje da se po ulasku u polje automatski otvara meni. Međutim, tu se javljaju nove dileme: šta ako je korisnik već izabrao odgovarajuću opciju iz menija, a onda ponovo pomeri kursor na to polje i slično.

Kao osnova za vašu funkciju koja automatski otvara meni može vam poslužiti funkcija **Look** sa slike 1. Ona se koristi kad tekuća promenljiva uzima vrednosti iz nekog niza. Na primer, može se pozvati sa:

```
@ 1,1 GET cVal WHEN Look(acValues)
@ 2,1 GET x WHEN Look( {"1", "3", "5"} )
```

gde je **cVal** promenljiva čija je vrednost jedan od elemenata iz niza **acValues**. Kad korisnik dođe na polje za unos promenljive **cVal**, otvara se meni iz koga se bira jedan od elemenata iz niza **acValues**.

Funkcija **Look** koristi objekte **GET** sistema na sledeći način: kada se aktivira funkcija, prvo se pomoću funkcije **GetActive** dobija tekući **GET** objekat. To je u stvari slog koji sadrži sva potrebna informacije o promenljivoj čiju vrednost treba uneti (za detaljan opis pogledajte funkcije **GET** sistema). Zatim se aktivira objekat – pošto se funkcija **Look** izvršava kao uslov koji treba ispuniti za unos promenljive, **GET** objekat još nije aktiviran. Kada se pošalje poruka **setFocus()** inicijalizuje se bafer za unos vrednosti, pozicija kursora i slično. Za nas je najvažnija vrednost **oGet:buffer** – to je string koji sadrži vrednost promenljive bez obzira na tip. To nam omogućava da istu proceduru koristimo za promenljive **proizvoljnog** tipa – spisak opcija **uvek** treba pisati kao stringove. Kada korisnik izabere vrednost (funkcija **aChoice**) u bafer se upisuje izabrana vrednost i porukom **Assign()** nalaže se da se vrednost iz bafera dodeli promenljivoj.

Zavisne GET promenljive

... će dosta ćesto ... da se u okviru ... od unosa pojavljuju zavisne promenljive. Na primer, želite da omogućite korisniku da unese vrednost u dinarima ili u devizama. Ako korisnik unese vrednost u dinarima, polje u kome je devizna vrednost treba automatski da promeni vrednost i obrnuto. Problem se javlja sa prikazivanjem promenjene vrednosti dok se ne zada druga čija **GET** sistem ne manja spisane vrednosti na ekranu. Ovaj problem se može rešiti na više načina, a najjednostavniji je sledeći. Kod svake promenljive koja je zavisna sa nekom drugom stavlja se opcija **VALID** koja računa ostale vrednosti i ispisuje sve promenjene vrednosti. Na primer:

```
@ x,y GET Dinara
    VALID {|| Maraka := Dinara / VredMar,
              GetList[3]:display(), .T.}
```

Kada se bude unela vrednost promenljive **Dinara**, potrebno je izmeniti vrednost promenljive **Maraka**. Po pritisku na <Enter> izvršava se **VALID** izraz koji računa novu vrednost promenljive **Maraka**. Međutim, to nije dovoljno. Na ekranu će ostati stara vrednost te promenljive sve dok se ne ispiše nova. To ispisivanje će se automatski obaviti kad se kursor pomeri na promenljivu **Maraka**, pa su mnoga rešenja ovog problema zasnovana na nekim kombinacijama **KEYBOARD** komande. Mnogo je elegantnije iskoristiti to što je svaka promenljiva predstavljena objektom u nizu **GetList**. Kada treba ispisati neku od vrednosti, dovoljno je poslati poruku **display** odgovarajućem objektu. Za naš primer, ako je promenljiva **Maraka** specificirana treće, treba izvršiti **GetList[3]:display()**. Na kraju treba vratiti vrednost **.T.**, što označava da je upravo uneta vrednost korektna. Naravno, sve ovo se može izvesti i korišćenjem običnih funkcija, a ne kodnih blokova.

Na kraju jedna napomena. Da li ste znali da je moguće izmeniti **GET** sistem bez prepravke datoteke **GETSYS.PRG**? U spisku internih promenljivih svakog **GET** objekta nalazi se i promenljiva **reader** koja sadrži kodni blok zadužen za unos vrednosti. Izmenom ove promenljive možete preusmeriti unos na neku vašu funkciju. Čak možete u okviru jedne **READ** naredbe kombinovati originalni *Clipper*-ov i neki vaš sistem za unos podataka.

> Ova rubrika postoji zbog vas – šaljite svoje priloge, kritikujte, hvalite ili jednostavno kažite svoje mišljenje.

Listing 1

```
*************************************************
* DATUM   5. 8.1991
*
* NAZIV   Look (Options)
*
* OPIS    Izbor elementa iz OPTIONS
*************************************************

PROC Look (Options)
   LOCAL ItemNo, n, Length, x, y, oGet

   // Koji se GET unosi?
   oGet := GetActive ()

   // Aktivira tekući GET
   oGet:setFocus()

   // ASCII vrednost tekućeg GET-a
   Value := oGet:buffer

   // Opcije
   n := Len(Options)
   Length := Len(Options[1]) + 2
   ItemNo := AScan (Options, Upper(Value))

   // Brisanje i crtanje okvira
   x := Row()  ; y := Col()
   cScr := SaveScreen (X,Y,X+N+2,Y+Length+1)
   @ X,Y CLEAR TO X+N+2,Y+Length+1
   @ X,Y TO X+N+2,Y+Length+1

   // Izbor opcije
   SET CURSOR OFF
   ItemNo := AChoice(x+1, y+1, x+n, y+Length,;
         Options, .T., , ItemNo, ItemNo-1)
   RestScreen (X,Y,X+N+2,Y+Length+1, cScr)
   SET CURSOR ON

   // Ako je korisnik izabrao vrednost, upiši je
   IF ItemNo <> 0
      oGet:buffer := Options[ItemNo]
      oGet:Assign ()
   END IF

RETURN .F.
```

Funkcija za unos vrednosti *GET* promenljive iz menija

Zoran Životić

FOR

C **for** petlja je jedan od zaštitnih znakova ovog jezika. Razlog češće upotrebe **for** petlje nego što bi se moglo očekivati s obzirom da postoje još dve kontrolne strukture ovog tipa (**do** i **while**) je pre svega njena praktičnost – **for** u jednom potezu objedinjuje inicijalizacioni deo petlje, ispitivanje uslova za izlazak i pripremu za sledeću iteraciju. Praktično to izgleda ovako:

```
for( izraz1; izraz2; izraz3 ) {
      naredbe...
   }
```

a interno se prevodi u sledeći, drugim jezicima poznatiji oblik:

```
izraz1;
   while ( izraz2 ) {
      naredbe...
      izraz3;
   }
```

Teorijski, nema nikakve razlike između ova dva oblika. **For** je samo pregledniji način pisanja, jer prisustvo **izraza1** i **izraza3** funkcionalno ne znači ništa. Činjenica je, ipak, da se u praksi pre skoro svake petlje vrši neka inicijalizacija promenljivih – zato je daleko preglednije da se taj kod ubaci u samu **for** petlju. Ako se **izraz1** na ovaj način vadi iz ostatka programa i jasno stavlja tamo gde mu je funkcionalno mesto, onda je izdvajanje **izraza3** još značajnije sa stanovišta preglednosti. Obično je već na prvi pogled na **for** petlju jasno šta se u petlji dešava, pa se pri analizi tuđih programa najčešće i koncentrišemo na ova mesta.

Naravno, C ne bi bio C da ne omogućava i zloupotrebe. Svaki od tri dela **for** kontrole se može izostaviti, pa vas ništa ne sprečava da, recimo, **for** koristite na potpuno isti način kao i **while**:

```
izraz1;
   for( ; izraz2; ) {
      naredbe...
      izraz3;
   }
```

Na sreću, ovakva upotreba se retko sreće – čak i kada se izostavlja neki od tri izraza, to se radi sa određenim smislom. Tako, recimo, oblik

```
for( ; izraz2; izraz3)...
```

obično ukazuje da je petlja nastavak nekog dešavanja od ranije i da u prethodnim linijama koda treba potražiti objašnjenje. Izostavljanje **izraza3** najčešće ukazuje da petlja ne koristi brojač iteracija, dok se izostavljanje sva tri izraza koristi u situacijama kada nam je potrebna "beskonačna" petlja unutar koje postoji puno mogućih uslova za izlazak.
Ovaj poslednji slučaj se često zloupotrebljava. S obzirom da se **izraz2**, za razliku od **do/while** konstrukcije, ispituje **pre** svake iteracije, onda se obično napiše nešto poput:

```
for( ;; ) {
      naredbe
      if (izraz2)
         break;
   }
```

što svakako ne doprinosi preglednosti programa. Ipak, "beskonačna" **for** petlja se često sreće i veoma je korisna u složenijim slučajevima – dovoljno je, recimo, da imamo:

```
for(;;) {
      naredbe...
      if (izraz2.a) break;
      naredbe
      if (izraz2.b) break;
      ...
   }
```

pa da **for** predstavlja najjednostavnije rešenje. Naravno, podjednako bi se moglo upotrebiti, recimo

```
while (1) {
```

ali ovakav izraz će neke kompajlere naterati da vas upozore da je logički izraz u stvari konstanta, što je inače razložno upozorenje u svim situacijama sem ove.

Ima li ograničenja u pogledu složenosti i načina na koji se formiraju **izraz1,2,3**? Praktično nema – sve što se nalazi između znakova ; se tretira kao jedan odvojen izraz i na njega se primenjuju sva standardna pravila C-a. Tako, uz korišćenje zarez operatora, **for** petlja može pregledno da koristi i više promenljivih kao brojače, recimo:

```
for(i=0,j=MAX; i<MAX || j>=0; ++i,--j)
```

kao što se u izrazu mogu pojaviti i funkcije:

```
for( init(i,p) ; checkr(r) ; obn(r,p,a) )
```

Ipak, ne treba gubiti iz vida osnovnu prednost **for** kontrole toka – preglednost. Zato izraze treba formirati tako da vama (a time i drugima) najbolje predstave suštinu onoga što se dešava u petlji.

Beli razmaci...

Ovaj podnaslov, kao i slični bukvalni prevodi sa engleskog jezika, malo govori o onome što će biti predmet razgovora. U našem jeziku nemamo adekvatan izraz za **white-space** znake, što i nije čudno s obzirom da se radi o jednom, čak i za programere, retko korišćenom terminu. Ipak, oni koji se stalnu sa njim imaju šansu da ga zapamte za ceo život, naročito ako odluče da umesto uz literaturu znanje steknu praktičnim eksperimentisanjem. Šta su, dakle, **white-space** znaci?

Tri znaka se objedinjeno nazivaju **white-space** znacima: razmak (ascii kod 32), tabulator (kod 9) i novi red (kod 10). Kontekst u kome je ovo grupisanje bitno je vezan za **scanf** funkcije i to je možda i jedini slučaj gde se ova tri znaka tretiraju na isti način – zato je i sam termin slabo poznat.

Osnovna karakteristika **scanf** funkcije je da se podaci iz formatskog stringa koriste za poklapanje sa stringom koji se čita – ako napišemo recimo %**d** to znači da očekujemo da u njemu sledeći niz znakova čini celobrojnu vrednost, ako napišemo %**s** očekujemo da sledeći niz znakova formira string. Sve drugo što se napiše bez ovog specijalnog formatskog prefiksa % treba da se poklopi sa onim što se nalazi u nizu – u suprotnom **scanf** funkcije odustaje od dalje analize ulaznog niza. Ova karakteristika je toliko jasna i očigledna da se obično odustaje od daljeg čitanja dokumentacije. Primenom samo ovog znanja možemo doći u sledeću situaciju.

Umesto korišćenja **fgets** datoteku treba na sličan način liniju po liniju analizirati sa **fscanf** funkcijom. U datoteci se očekuju tri reda podataka, od kojih neki (ili svi) mogu biti i prazni, što za program takođe ima određen smisao. Prvi problem na koji se naiđe, o čemu je već bilo reči, je činjenica da %s formatska oznaka stringom smatra samo ono što se nalazi između razmaka – ako naša datoteka ima sledeći sadržaj (oznake **red** su samo zbog preglednosti, nema ih u datoteci):

```
1 v1a v1b
2 v2
3 v3
```

onda čitanje

```
fscanf(f,"%s\n%s\n%s",s1,s2,s3);
```

daje rezultat **s1=v1a**, **s2=v1b** i **s3=v2**, što nije ono što nam treba – naše **s1** bi trebalo da ima sadržaj **v1a v1b**. O rešenju smo već pisali, a sastoji se u upotrebi posebne formatske oznake %[^**xyz...**] koja će pročitati niz znakova sve do onog trenutka dok ne naiđe na neki od znakova iz skupa **xyz...**. To konkretno znači da se ceo red, bez obzira na razmake u njemu, može pročitati kao jedan string na sledeći način:

```
fscanf(f,"%[^\n]",s1);
```

Izmenimo tako naš primer na sledeći način:

```
fscanf(f,"%[^\n]\n%[^\n]\n%[^\n]",
  s1,s2,s3);
```

Linija se malo teško čita, ali je ideja da se umesto %**s** primeni %[ ] da se stigne do znaka **\n** (kraj reda), zatim da se navođenjem tog znaka kao sledećeg člana formatskog stringa i on preskoči i stigne na sledeći red i tako za sva tri parametra. Ipak, tek sad počinju problemi. Pretpostavimo da datoteka ima sledeći sadržaj:

```
1 v1a v1b
2
3 v3
```

Rezultat će ponovo biti neočekivan – **s1=v1a v1b**, **s2=v3** a **s3**... **s3** neće dobiti nikakav sadržaj jer će funkcija **fscanf** odustati i vratiti broj 2 kao oznaku da su samo 2 elementa formatskog stringa zadovoljena čitanjem.

Razlog je u bukvalnoj primeni pravila nad znakom **\n** – novi red. Pravilo u njegovom slučaju kaže da se podudarnost ostvaruje ne samo nad jednim znakom, već na **proizvoljnom broju uzastopnih** pojava ovog ili **bilo kog drugog white-space** znaka. To znači da je, nakon prve linije, onaj jedan znak **\n** u formatskom stringu naterao **fscanf** funkciju da pročita kraj reda linije 1, ali i kraj reda linije 2, jer je ona prazna, pa tako njen **\n** pada odmah iza **\n** znaka iz prethodne linije. Čak i ako liniju 2 napunimo razmacima sa namerom da ova dva znaka razdvojimo, desiće se potpuno isto, jer i razmak spada u **white-space**. **fscanf** ne pravi nikakvu razliku između tri **white-space** znaka, pa bi se potpuno isto desilo i kada bismo napisali

```
fscanf(f,"%[^\n] %[^\n] %[^\n]",
  s1,s2,s3);
```

jer će razmak imati isti efekat kao i **\n** (naravno, sve važi i za treći znak, tabulator). Pravilo bukvalno kaže "proizvoljan broj uzastopnih", što znači i nijedan. Dakle, potpuno je svejedno da li pišemo

```
fscanf(f,"%[^\n] %[...                     ili

fscanf(f,"%[^\n] %[...                     ili

fscanf(f,"%[^\n] \n \n\n \n %[...
```

White-space znaci su dakle izuzetak iz opšteg pravila koja važi u **scanf** funkcijama. To naravno nije slučajno urađeno tako – ovakvo ponašanje funkcija je više nego korisno jer omogućava da ulazni niz/datoteku tretiramo kao neprekidni niz znakova u kome podela po redovima nema posebno značenje. U slučajevima kada nam je ova podela važna, najjednostavnije je skoro obavezno koristiti **fgets/sscanf** kombinaciju. Problem nije samo u tome da se pročita **jedan \n**, to se da rešiti na sledeći način

ih u datoteci)

```
fscanf(f,"%[^\n]%*c%[^\n]%*c%[^\n]",
  s1,s2,s3);
```

Ovde će %*c pročitati jedan znak (koji god da je), a oznaka * znači da pročitano polje **ne treba dodeliti** promenljivoj (u listi parametara se, dakle, ne očekuje pointer na promenljivu – ovo važi i za sve ostale tipove ulaznih polja). Međutim, čak i ovako napisana **fscanf** će pasti u slučaju da je, recimo, drugi red prazan. Ovde već zalazimo u oblast koja je toliko specifična da nije ni dokumentovana, pa nisam siguran da li je ovo opšte pravilo. Naime, ako format %[^\n] dođe u situaciju da treba da pročita prazan string, dakle da je prvi znak upravo onaj koji prekida dalje čitanje, **fscanf** funkcija prekida rad. Verovatno je opšte pravilo da prekid nastupa kada se zadati format ne može primeniti na ulazni niz, ovde interpretirano tako da se prazan string ne računa u važeći podatak, što i nije bez razona.

Bilo kako bilo, **scanf** ostaje i dalje jedna od najkorisnijih bibliotečkih funkcija – jedna od onih koju što više koristite sve više otkrivate njenu vrednost. Ipak, treba imati na umu generalno pravilo po kome ova funkcija tretira ulazni niz isključivo kao **niz polja** – podela ulazne datoteke na linije nema poseban smisao.

Vladimir Stamenović

Svim korisnicima Lotusa *1-2-3* je poznato, ili bi trebalo da bude, da je jedna od najjačih strana ovog programa mogucnost kreiranja atraktivnih grafikona na bazi podataka iz tabele.

IZ TABELE U GRAFIKON

Karakteristično je da se u jednoj tabeli na osnovu raspoloživih podataka, može napraviti više grafikona različitih tipova. Nakon kreiranja jednog grafikona komandom **/Graph Name Create** dacete ime tom grafiku, a zatim napraviti novi itd. Dakle bitna stvar je da prvo uradite sva podešavanja da dobijete grafikon, pa tek onda mu date ime. Kada imate više grafikona, onda možete komandom **/Graph Name Use** reći programu koji vas grafikon interesuje (po imenu) i pogledati ga.

Kad je već tako, onda nije nikakav problem napraviti makro proceduru koja će prikazivati seriju vaših grafikona jedan za drugim. Najjednostavniji način da se to uradi je kao na slici 2. Ukratko, makro komande {WINDOWSOFF} i {PANELOFF} omogucuju da se grafici prikazuju u nizu bez vraćanja u tabelu i karakterističnog blinkovanja linije komandnog menija. Praktično, zamrzavaju sve osim grafičkog prozora. U drugoj liniji procedure pozivamo **MENU** (/) **Graph** (G) **Name** (N) **Use** (U) ime prvog grafika (gr1) i na kraju ENTER (~). Nakon toga dobijamo na monitoru prvi grafikon (gr1). Pritiskom na bilo koji taster, slika nestaje i makro procedura se nastavlja u trećoj liniji **Name** (N) **Use** (U) ime sledećeg grafika (gr2) i ENTER (~). Na kraju procedure, u poslednjoj liniji su komanda QUIT (Q) za izlaz iz **Graph** menija i makro komanda {HOME} da bi nakon izvršenja procedure pointer bio u tabeli u ćeliji A1.

Ovakvom makro procedurom postižemo mogucnost pregleda grafički prikazanih podataka po unapred utvrđenom redosledu. Imamo dovoljno vremena da pregledamo svaki grafikon, jer on ostaje na ekranu sve dok ne pritisnemo bilo koji taster za nastavak procedure. Može se to napraviti i tako da nakon poslednjeg grafikona ide ponovo prvi, sve dok se ne pritisne neki određeni taster, na primer Esc.

Ovaj način prikazivanja podataka je koristan u raznim prilikama, poslovnim susretima i slično. Međutim, ako vašeg poslovnog partnera interesuje neki grafikon koji je prošao u pregledu, onda mora da prođe i sve ostale grafike da bi završio makro proceduru, da je startuje ponovo i pregleda dok ne dođe do onoga koji ga interesuje. Priznaćete da onda korisnik dosta gubi na komfornosti.

Da razmotrimo drugu mogućnost. Na primer, želimo da napravimo makro proceduru koja će omogućiti korisniku da se u toku pregleda grafikona kreće i unapred i unazad, odnosno da može da se vrati na prethodni grafikon koji je već video, pa onda da nastavi dalje na naredni, sve dok ne zaželi prekid, a njega omogucuje pritisak na određeni taster, na primer k ili K (od – kraj).

Ako hoćete da probate kako to radi, unesite tekst sa slike 3. Koristite komandu **/Range Name Labels Right** da nazivima iz kolone Q pridružite susedne ćelije iz kolone R. Nakon toga upotrebite komandu **/Range Name Create** da imenu „deo" pridružite ćelije R2..R3. Obratite pažnju na ćeliju R3, koju smo nazvali „index". U nju treba uneti formulu @INDEX(lista 0 broj).

Pre nego što pokrenete makro proceduru morate napraviti i imenovati nekoliko grafikona. Ja sam za ovaj slučaj koristio tabelu sa slike 1 (štampana je direktno iz *1-2-3* kroz WYSIWYG – *spreadsheet publishing*). Primera radi napravio sam šest grafikona raznih tipova koji prikazuju kretanje cena proizvoda (stavki 1-3) po mesecima (mesec 1-6). Nazvao sam ih sa GR1 do GR6. Lista imena grafikona je nazvana „lista" i nalazi se u oblasti ćelija Z3..Z8, kao što se vidi na slici 4. Pregled oblasti i njihovih imena korišćenih u ovom primeru, dat je na slici 5. Za njihovo prikazivanje upotrebljena je komanda **/Range Name Table**.

Da biste pokrenuli proceduru pritisnite MAKRO-G, odnosno na većini računara Alt-G. *Lotus 1-2-3* će prikazati prvi grafikon po redu. Ako pritisnete taster N ili n, dva puta, dobićete naredni grafikon. Ako pritisnete taster P ili p biće prikazan prethodni grafik. Makro procedura će biti u toku „radiće" sve dok ne pritisnete dva puta tastere K k (kraj) ili ENTER. Zašto tastere morate pritisnuti dva puta? Zato što prvi put sklanjate tekući grafikon sa ekrana, a drugi put kažete makro proceduri koji će grafikon prikazati kao sledeći.

Ključna formula je u ćeliji koju smo nazvali „index". Ta formula je @INDEX funkcija koja daje ime grafikona iz liste istih. Kada je vrednost ćelije „broj" jednaka 0 (nula), @INDEX funkcija daje ime prvog grafikona iz liste, u ovom primeru GR1, itd. Makro procedura kontroliše ovaj proces preko vrednosti u ćeliji „broj", a u zavisnosti od toga koji taster je pritisnut.

Evo bližih objašnjenja pojedinih linija u makro proceduri.

{**LET broj,0**}/G – smešta vrednost 0 u ćeliju „broj", aktivira slash (/) i poziva komandu **Graph** iz menija.

{**RECALC deo**} – preračunava oblast ćelija nazvanu „deo" da bi bila ažurirana formula u ćeliji „index" koja zavisi od vrednosti u ćeliji „broj".

NU index~{**GET taster**} – aktivira komandu **Name Use** iz **Graph** menija, poziva „index" kao potprogram, aktivira ENTER, a onda čeka dok mi ne pritisnemo taster. U ćeliji „index" je rezultat formule ime grafikona, koji se pojavljuje na ekranu i procedura pravi pauzu. Kada želimo da vidimo neki drugi grafikon, pritisnemo taster N ili P dva puta. Prvi pritisak briše sliku sa ekrana, a drugi se smešta u ćeliju nazvanu „taster".

{**IF taster="P"**}~**LET broj,broj-1**} – smanjuje vrednost u ćeliji „broj" za jedan, ako je pritisnut taster P ili p (prethodni).

{**IF taster="N"**}~**LET broj,broj+1**} – povećava vrednost broja u ćeliji „broj" za jedan, ako je aktiviran taster N ili n (naredni).

{**IF broj<0**}~**LET broj,**@ROWS(lista)-1} – daje vrednost „broj"-a za jedan manje od broja redova u „lista"-i, ako je vrednost „broj" već manja od nule. Ovo omogućuje da „broj" ne bude negativan, odnosno, da prikaže poslednji grafikon nakon prvog, kada smo aktivirali taster P ili p.

{**IF broj>**@ROWS(lista)-1}~**LET broj,0**} – daje vrednost 0 (nula), ako je „broj" već veći za jedan od ukupnog broja redova u „lista"-i. To nam omogućuje da vidimo prvi grafikon kada smo pritisnuli taster N ili n, a upravo smo gledali u poslednji sa liste grafikona.

Ova i prethodna linija u proceduri omogućuju kretanje napred-nazad po raspoloživim grafikonima.

{**IF taster="K"#OR#taster="~"**}{**ESC 2**}**QUIT**, – omogućuje prekid prezentacije grafikona, ako pritisnete dva puta tastere K k ili ENTER. U tom slučaju, aktivira se taster Esc dva puta i komanda QUIT završava makro proceduru. Nakon toga, nalazite se ponovo u tabeli.

{**BRANCH petlja**} – vraća tok makro procedure ponovo na ćeliju nazvanu „petlja". Tada komanda {RECALC} ažurira formulu u „index"-u baziranu na novoj vrednosti „broj"-a. Onda makro procedura prikazuje novi grafikon. Procedura pauzira dok ne pregledate grafik itd. itd.

Ovu makro proceduru možete primeniti u *Lotusu 1-2-3* verzija *2.x* i *3.x*, a sa malim izmenama i *Symphony*.

SOPSTVENI MENIJI

Ako ne koristite mogućnost kreiranja sopstvenih menija onda je to prava šteta, jer oni toliko skraćuju posao i štede vreme da je to za posebnu priču. Korisnici *Lotusa 1-2-3* i *Symphony* će se vrlo brzo, ako probaju, uveriti u to. No, nije mi namera da vas ubeđujem, morate probati pa prosuditi. Hteo bih da vam skrenem pažnju na još neke mogućnosti vezane za pravljenje sopstvenih menija. Naime, iskusni korisnici to znaju, a početnici će se uveriti da u svoj meni mogu da uvrste maksimalno osam (8) komandi. Pošto svaki meni mora, odnosno bar bi trebalo da ima i komandu za izlaz iz menija, to znači 7 komandi i jedna za izlaz. Tačno je da, zbog jasnoće i preglednosti, treba praviti manje menije, ali možda se nađete u situaciji kada vam je neophodno više opcija u meniju. Još kada je to „više" više od 8, eto problema. No, „Lotus" ne bi bio ono što jeste kada i za to ne bi bilo rešenja. Izuzetno korisna mogućnost pri pravljenju makro procedura u „Lotus"-ovim programima je to što se one mogu grupisati u rutine i pozivati u programu gde god je to potrebno. Elem, ako koristimo listu rutina kao meni, onda nismo ograničeni brojem opcija.

Kako bi to izgledalo pokazaću vam na jednom jednostavnom primeru. Pretpostavimo da imamo tabelu kao na slici 6. Ako u takvoj tabeli treba da uradimo više poslova kao što su unos podataka, štampanje delova ili cele tabele, snimanje tabele na disk itd. onda će biti logično da za te poslove napravimo svoj meni iz koga ćemo delovati.

Naš meni će imati 10 komandi, a po ovom sistemu možete napraviti meni sa neograničenim brojem opcija. Na slici 7 vidite kako izgleda meni. Sa leve strane su, odozgo na dole, poredane komande, a desno od njih su objašnjenja šta one rade. Komande iz ovog menija su u stvari nazivi rutina, a kako one izgledaju pogledajte na slici 8. Sva imena oblasti koja su korišćena i njihove adrese dati su u tabeli na slici 9.

Da vidimo kako to funkcioniše. Na slici 8 prva po redu je glavna makro procedura koja se zove \m, a poziva sa Alt-M. Ona prvo postavi pointer na početnu poziciju u tabeli (HOME), a zatim postavlja globalnu zaštitu na celu tabelu. Podrazumeva se da ćete skinuti zaštitu sa onih ćelija gde dozvoljavate unos podataka, a u ovom primeru sa ćelije u koloni Mesec1. Sledeća linija u glavnoj makro proceduri postavlja pointer na početak oblasti sa prikazom glavnog menija (GLMENI), a zatim na prvu opciju u meniju (oblast KOMANDE). Na kraju te linije imamo makro komandu {?} koja dozvoljava da izvršimo neku akciju pre nego što se nastavi dalje izvršavanje glavne makro procedure. U ovom primeru to nam omogucuje da strelicama na dole i gore postavimo pointer na odgovarajuću komandu rutinu.

Sledeći red (treći) je ključ našeg problema. Naime, izraz {DISPATCH @CELLPOINTER(„CONTENTS")} usmerava dalji razvoj do-

Slika 1



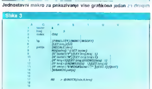
Tabela sa podacima na osnovu kojih su kreirani grafikoni

Slika 2



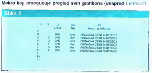
Jednostavni makro za prikazivanje više grafikona jedan za drugim

Slika 3



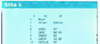
Makro koji omogućuje pregled svih grafikona (unapred i unazad)

Slika 2



Lista sa nazivima korišćenih grafikona

Slika 5



Pregled korišćenih oblasti i imena

gađaja u zavisnosti od toga šta pročita u ćeliji na kojoj se nalazi pointer. Da pojasnim. Ako nam je komanda ?} dozvolila da pomerimo pointer na odgovarajuću komandu rutinu, onda kada pritisnemo [ENTER] izraz DISPATCH @CELLPOINTER("CONTENTS") pročita sadržaj ćelije koju smo odabrali. Pošto je sadržaj te ćelije u stvari naziv rutine, onda se ista aktivira i tako nastavi izvršavanje cele procedure.

Ako u meniju odaberete opciju AKCIJA1 onda ćete imati mogućnost da unosite nove podatke u kolonu MESEC1 za sve stavke. Nakon unosa vraćate se ponovo u glavni meni. Ostale komande (Akcija 3-6) će samo aktivirati zvučni signal {BEEP}, tek da znate da je komanda izvršena, i prekinuće izvršavanje procedure. Za šta bi one mogle da se iskoriste ostavljam vama na volju.

Komanda AKCIJA7 će odštampati oblast STAMP1, odnosno kolone STAVKA i MESEC1. Nakon štampanja vratiće vas ponovo u glavni meni.

Komanda SNIMI će snimiti labelu na disk pod imenom "MENIPR". Nakon prvog snimanja imaćete jedno "r" (Replace) viška na kontrolnom panelu. Sa [Esc] ga obrišite. Kod drugog i daljih snimanja sve će biti u redu. Nakon snimanja vraćate se ponovo u glavni meni.

Komanda KRAJ služi za kraj rada sa labelom i vraća vas u DOS, odnosno Access meni.

Slika 6

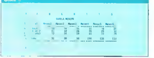


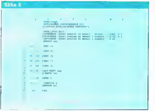
Tabela za koju je pravljen meni

Slika 7



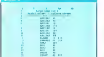
Izgled glavnog menija

Slika 8



Makroi/rutina koje čine glavni meni

Slika 9



Nazivi oblasti i adrese koje su korišćene u ovom primeru

Nadam se da će vas ovi primeri više zainteresovati za programiranje u *1-2-3* ili *Symphony*. U svakom slučaju bilo bi dobro da se javite redakciji i da predložite neku zanimljivu temu iz ove oblasti, a mi ćemo se potruditi da pripremimo još neki interesantan primer.

Nerado se članak započinje ispravkama prethodnih, ali... U pretposlednjem nastavku „Savetnika" („Računari" 79) potkrala se greška pri pripremi za štampu, tako da su potpisi ispod listinga programa R_FRACTAL i R_FILL međusobno zamenjeni. Nista strašno, zar ne – ali se to na žalost ne može reći i za omašku u drugom nastavku („Računari" 77), kada je ispao prvi red iz oba listinga, jer se radilo o programima u kojima on nije nevažan. Na početku prvog listinga treba da stoji **unit DATETIME**, a drugog **program CLOCK**. Naziv source datoteke jedinice (ove ili bilo koje druge) nije proizvoljan, mora se poklapati sa nazivom iza **unit**, inače program koji je uključuje neće moći da je prepozna nakon što se ova prevede u TPU datoteku.

Kodovi tastera = ?

Preovladava mišljenje da u *Turbo Pascal*-u očitavanje tastature nije rešeno na najbolji način. Postoje svega dve funkcije namenjene toj svrsi: **KeyPressed** vraća **True** ako je bilo šta pritisnuto, a **ReadKey** vraća kao **char** konkretan kod tog tastera. Dobro, sličan par komandi postoji i u većini drugih popularnih jezika i njihovih prevodilaca.

Problem je u tome što su Borlandovci zamislili da one dejstvuju vezano: da se prvo sa **KeyPressed** sačeka trenutak pritiska na neki taster, a odmah iza toga da se putem **ReadKey** utvrdi njegov kod. Definisano je, međutim, da **ReadKey**, ako **KeyPressed** ima vrednost **False**, čeka na pritisak – čime se praktično zaustavlja izvršavanje programa, što je neprihvatljivo ako dotična funkcija čini deo petlje u kojoj se ispituje stanje više od jednog tastera. Još gore od toga, sa **ReadKey** nismo u stanju da registrujemo signale sa dva različita tastera u toku istog sistemskog interapta. A kako interapti kod PC računara nemaju veliku učestalost, dešava se da naš program kasni u reakciji, pa čak i da previdi poneki pritisak.

Izlaz iz svega ovog je dvojak: ili da preko poziva BIOS-u podesimo trajanje pauze do početka automatskog ponavljanja i brzinu tog ponavljanja, ili da jednostavno napišemo sopstvene funkcije za očitavanje tastature – prvo rešenje je opštepoznato, a drugo trivijalno, tako da ih nećemo ovom prilikom izlagati. Korisnije bi bilo da objasnimo kako se dolazi do kodova tastera koje treba testirati u programu: oni nam stalno trebaju, a broj raznih kombinacija sa i bez Ctrl i Alt je toliki da ih ne možemo znati napamet, a slabo bi pomogla i neka dugačka tablica. Moguće rešenje je zato dato na listingu 1. Radi se o vrlo korisnom programu KEY_NUM (sličan je sebi sačinila većina iskusnijih programera) koji trenutno ispisuje kod željene tipke.

Iz programa se izlazi sa Control-Break, a da bi se to osiguralo, postavlja se **CheckBreak** na **tačno**. Primetite da specijalne tastere prepoznajemo prvo po nuli, pa tek u sledećem čitanju sa **ReadKey** po karakterističnom kodu. U dosadašnjim primerima smo često očitavali tastaturu, tako da bi po tom pitanju trebalo sve da bude jasno. Što se tiče tipki koje ne možemo ispitivati sa **ReadKey**, a to su **CapsLock**, **NumLock**, **Insert** i slično, očitavanje njihovog stanja vršimo direktno iz BIOS segmenta, tačnije iz pojedinih bitova sa adresa $0017 i $0096.

Integrisani asembler

Novosti u verziji 6.0 su kvalitetniji editor, skup objekata namenjen izradi korisničkih interfejsa (**TurboVision**) i integrisani asemblerski prevodilac – ovo poslednje je, ne slučajno, najviše obradovalo korisnike. Naime, do sada su postojala dva relativno nekomforna metoda povezivanja *Turbo Pascal*-a sa funkcijama i procedurama pisanim u asembleru. Prvi je podrazumevao umetanje nekoliko kodova mašinskih instrukcija unutar listinga kroz **inline** deklaracije, a drugi putem deklaracije **external** rutine u asembleru napisane u TASM-u prema određenim pravilima i prevedene u .OBJ datoteke linkovane su sa programima u *Turbo Pascal*-u.

Pojavom verzije 6.0 obe ove metode praktično gube značaj, jer je u okviru istog listinga omogućeno ne samo pisanje zasebnih procedura u asemblerskom jeziku i pascalu, nego i mešovitih, po sistemu „malo ovo, malo ono". Glavne odlike integrisanog asemblera su i mogućnost generisanja koda za 8086, 8087, 80286 i 80287 procesore i koprocesore, string konstante, specijalni tipovi **dword**, **qword** i **tbyte**, labele bez definisanja unapred, prefiksi instrukcijama, automatsko umetanje ulaznog i izlaznog koda i posebno rezervisana varijabla **Result** za preuzimanje rezultata funkcija. Uz to, asembler je potpuno integrisan sa celim paketom, znači ne samo sa pascal kompajlerom nego i sa dibagerom: naše rutine možemo izvršavati korak po korak, postavljati prekidne tačke, pratiti i menjati stanje registara i steka itd. Iako asemblerski prevodilac po mogućnostima nije na nivou MASM-a, pa ni TASM-a, a ni dibager na nivou *CodeView*-a, teško da će nam neka opcija koju ne poseduju stvarno i zatrebati, mašinsko programiranje u okruženju *Turbo Pascal*-a se dosta razlikuje od onog na koje smo dosad navikli.

Praksa je pokazala da se povezivanju TP-a sa asemblerom pribegavalo najčešće kada je trebalo pozivati DOS i BIOS interapte, efikasnije upravljati **heap** memorijom, operisati sa dugačkim stringovima, slati poruke periferním uređajima i naročito pisati brze grafičke rutine. Ključna je nova rezervisana reč **asm** koja znači isto što i **begin**, samo što iza nje slede instrukcije u asembleru. Ako je procedura ili funkcija koju nameravate da pišete sva u asembleru, treba staviti iza zaglavlja i reč **assembler**. Lokalne konstante i promenljive mogu se deklarisati i unutar rutina, ali i na potpuno uobičajen način preko **var**. Labele se koriste za označavanje mesta na koje treba skočiti, ali cini se da, na žalost, ne mogu da posluže kao pokazivači na adresu sa koje želimo pročitati ili na koju želimo upisati vrednost. (Ovo svojstvo bi bilo posebno vredno pri pisanju rutina koje menjaju sopstveni kod.)

Osnovna pogodnost mašinskog programiranja u okruženju *Turbo Pascal*-a 6.0 (jasno, i nove verzije *Turbo C*-a) je da se možemo direktno pozivati na većinu njegovih konstanti, varijabli, procedura i funkcija bez ikakvih ranijih EXTRN definicija i sličnoga. Uključujući tu i naše potprograme u paskalu, bez obzira da li su unutar istog programa ili modula, važno je samo da su [illegible] u listingu ili da su deklarisani sa **forward**. Uvek je pravilo kod pozivanja drugih potprograma da se prvo na stek stave registri i promenljive koje želimo sačuvati, zatim svi argumenti, istim redom kako su navedeni u listi formalnih parametara, da se sa CALL instrukcijom izvrši poziv i, na posletku, vrate potrebni registri i promenljive u predašnje stanje. Ako smo pozvali neku od funkcija, rezultat će nam biti vraćen kroz registre na način koji zavisi od tipa podatka. Boolean vrednost logičke funkcije vraćće se u AX tako da je 0 – False, 1 – True. Svi celobrojni tipovi dužine dva bajta će se isto tako nalaziti u AX, a oni od jednog bajta samo u AL. **LongInt** će biti u paru DX:AX. Pointeri takođe: segment u DX, ofset u AX. Tip **real** se vraća u DX:BX:AX, dok je kod stringova za nijansu složenije: treba pre poziva na stek staviti, pre svih ostalih parametara, pokazivač na početnu adresu memorijskog prostora gde želite da vam string funkcija upiše rezultat. Ovih pobrojanih pravila morate se pridržavati i kada vaša asemblerska rutina prihvata rezultat i kada ga prosleđuje ostatku programa u paskalu.

Rekurzija

Pošto, dakle, naša rutina u asembleru može jednostavno da poziva druge funkcije i procedure, postavljava se pitanje da li može da poziva i samu sebe? Može, i to na dva načina: prvi podrazumeva slanje argumenata putem steka, a drugi putem registara. U prvom pristupu je postupak, kao gore, identičan pozivanju nekog drugog potprograma, tj. izvodi se sa **CALL ImeProc**. Ako imamo visok stepen rekurzivnosti, obavezno treba direktivom $F- lokalno definisati pozive tipa NEAR. Nakon testiranja možemo izvršavanje ubrzati direktivom **$S-**, kojom preskačemo proveru prekoračenja steka. U drugom pristupu se rekurzivni poziv vrši tako što se na stek sklone važni registri i/ili promenljive, a potom se sa **CALL ImeLabele** ide na adresu u rutini označenu dotičnom labelom. Ovo je brži način, jer se preskače obimni ulazni kod koji generiše kompajler kao zaglavlje funkcijama i procedurama, ali ovako nismo u stanju da kontrolišemo slaganje na stek, pa ako dođe do njegovog prepunjavanja, garantovano nam pada ceo sistem.

Kod pristupa podacima složene strukture (matrice, slogovi, objekti) deklarisanim u paskalu, osnovno je znati njihovu početnu adresu u memoriji. Tu adresu rutini u asembleru možemo proslediti ili preko kompletnog pointera ili samo preko ofseta funkcijom **Ofs(Varijabla)**. Drugi način se koristi za one podatke za koje sigurno znamo da će biti u DATA segmentu, u tom slučaju je sve vrlo prosto, jer nema potrebe menjati vrednost DS registru, a adresu dobijamo kompaktnu kroz ulaznu promenljivu tipa **word**. Pažnju valja obratiti i na format pakovanja podataka u memoriji – on je često upravo suprotan od očekivanog, tako na primer sledeći član neke tabele nije onaj desno, nego onaj dole. Pošto ovde nekih strogih pravila skoro da i nema (odnosno, veoma su komplikovana), najlakše ćete format pakovanja dokučiti pomoću **Debug/Evaluate** prozora, ispitujući ofset adrese jednog po jednog elementa strukture.

Popunjavanje uzorkom

Smatrali smo da je najpogodnije dati asemblersku verziju neke već ranije objavljene rutine, u cilju da se čitalac što manje zadržava na samom algoritmu, već da svu pažnju usmeri na razumevanje gore opisanih tehnika. Na listingu 2 se, prema tome, nalazi program ASM_FILL, sličan programu R_FILL iz pretposlednjeg izdanja rubrike, s tim razlikama da se sada ispunjavanje zatvorene konture ne vrši

Listing 1 KEY_NUM

```
program KEY_NUM; { @ Dlanjin 1991 }

uses    Crt;

var     Taster : char;

begin
  CheckBreak:=True;
  repeat
    Write('Kod tastera je: ');
    repeat until KeyPressed;
    Taster :=ReadKey;
    if Taster = #00 then
      WriteLn('00H # ',Ord(ReadKey))
    else
      Writeln('#',Ord(Taster));
  until false
end
```

Program koji ispisuje kod željene tipke

Listing 2 ASM_FILL

```
program ASM_FILL;       { Primer upotrebe asemblera u Turbo Pascalu  }
                        { Miroslav Dlanjin (c), Januar/Februar 1991  }

uses    Crt, Graph;

const   Uzorak   array [0..7, 0..7] of word =  ((1,0,0,0,0,0,0,0),
                                                (1,1,0,0,0,0,0,1),
                                                (0,1,1,0,0,0,1,0),
                                                (0,0,1,1,0,1,0,0),
                                                (0,0,0,1,1,0,0,0),
                                                (0,0,0,1,1,1,0,0),
                                                (0,0,1,0,0,1,1,0),
                                                (0,1,0,0,0,0,1,1));

{$G+}  { Generise se kod za procesor 80286 i kasnije ! }

procedure AsmFill (Px, Py, BojaPoz, UzorakPtr : word); assembler;

var             Brojac, Petlja : word;

asm             PUSH Px             { OX  asm  BX @start je inicira   }
                PUSH Py             { juci deo koji se izvršava samo  }
                CALL GetPixel       { jednom na početku. Prvo se isp  }
                CMP  AX,BojaPoz     { ituje da li je tačka od koje se }
                JNE  @exit2         { kreće u boji pozadine. Zatim se }
                PUSH $000F          { postavlja vrednost pomoćne boje }
                CALL SetColor       { popunjavanja. Koordinate x se   }
                MOV  BX,Px          { puni BX i ide se na rekurzivni  }
                MOV  Brojac,$0001   { deo rutine. Brojač steka je na  }
  @start:       MOV  CX,BX          { jedan. Koordinata x je i u BX i }
                PUSH CX             { u CX koja se uz pomoć pascalove }
  @left:        DEC  BX             { funkcije GetPixel dolazi do one }
                PUSH BX             { tačkice različite boje od boje  }
                PUSH BX             { pozadine. To je levi kraj lini  }
                PUSH Py             { je u pomoćnoj boji popunjavanja }
                CALL GetPixel       { Primetite da prva od PUSH ins   }
                POP  BX             { trukcija spašava koordinat reg  }
                CMP  AX,BojaPoz     { BX, a druga dva šalju parametre }
                JE   @left          { funkciji. Rezultat,boju piksela }
                INC  BX             { na datoj poziciji, dobijamo u   }
                POP  CX             { akumulatoru. Sve dok je on      }
                PUSH BX             { identičan BojiPoz umanjuje se x }
  @right        INC  CX             { koordinata u BX i ponavlja se   }
                PUSH CX             { petlja. Kako će BX na izlazu po }
                PUSH CX             { kazivati na prvi levi piksel ra }
                PUSH Py             { zličit od BojaPoz,to je neopho  }
                CALL GetPixel       { dno inkrementirati ga za jedan  }
                POP  CX             { A na sličnom principu se u      }
                CMP  AX,BojaPoz     { narednoj petlji nalazi krajnja  }
                JE   @right         { desna tačkica u boji popunjava- }
                DEC  CX             { nja. Sada je, dakle, u BX leva  }
                POP  BX             { x koordinata, u CX desna,a u Py }
                PUSHA               { y koordinata, pa možemo povući  }
                PUSH BX             { vodoravnu liniju i spojiti ih.  }
                PUSH Py             { Pozivamo Line proceduru a pre   }
                PUSH CX             { toga smo joj preko steka pri    }
                PUSH Py             { premili parametre. Sve vrednosti }
                CALL Line           { registara smo sačuvali parom    }
                POPA                { PUSHA-POPA. Kasnije će nam dva  }
                PUSH BX             { puta zatrebati BX, i stavljamo  }
                INC  Py             { ga na stek. Sledeći korak je is }
  @loop1:       CALL @recur         { pitivanje okoline iznad i ispod }
                INC  BX             { pomoćne linije.                 }
                CMP  BX,CX          { U ciklusima @loop1 i @loop2 tra- }
                JBE  @loop1         { ga se za tačkama neposredno is  }
                DEC  Py             { pod i iznad, redom od BX do CX. }
                DEC  Py             { Razlog zbog kojeg postoje dve   }
                POP  BX             { povezane petlje umesto jedne je }
                PUSH BX             { sledeći: nakon sto je izvršen   }
  @loop2:       CALL @recur         { povratak iz ntže pozvane rutine }
                INC  BX             { treba korigovati brojač BX tako }
                CMP  BX,CX          { da se preskoči ispitivanje onog }
                JBE  @loop2         { segmenta linije BX-CX kojeg je  }
                POP  BX             { već ispitala i iscrtala niže    }
                INC  Py             { pozvana rutina. Tako se iznego  }
  @loop3:       PUSH BX             { va popunjavanja piksela koji su }
                PUSH CX             { iste boje kao i BojaPoz, ali    }
                PUSH BX             { koji u stvari pripadaju uzorku. }
                PUSH Py             { Nakon što su se @start i @loop2 }
                MOV  CX,Py          { izvršile, čime je osigurano da  }
                AND  BX,$0007       { iznad i ispod pravca BX-CX nema }
                AND  CX,$0007       { slobodnih pozicija, iscrtava se }
                SHL  BX,4           { niz tačkica prema definiciji iz }
                SHL  CX,1           { uzorka, a preko ranije povučene }
                ADD  BX,CX          { bele pomoćne linije. Da bi se u }
                ADD  BX,UzorakPtr   { matrici uzorka pronašao odgova  }
                MOV  AX,[BX]        { rajući broj, treba prvo iz x i  }
                PUSH AX             { iz y koordinate izdvojiti tri   }
                CALL PutPixel       { poslednja bita, jer matrica je  }
                POP  CX             { dimenzija 8x8. x pomnožiti sa   }
                POP  BX             { 16 i dodati mu dvostruki y te   }
                INC  BX             { ofset početka definicije uzorka }
                CMP  BX,CX          { u DATA segmentu. Nakon toga iz  }
                JBE  @loop3         { memorije učitamo tu vrednost u  }
                DEC  Brojac         { akumulator, njega postavimo na  }
                JNZ  @backto        { stek, iza svih ostalih parame   }
                JMP  @exit2         { tara, te pozivamo PutPixel. Ovo }
  @recur:       PUSH CX             { se sve dešava u @loop3, BX-u BX }
                PUSH BX             { kojeg inkrementiramo na dužini  }
                PUSH BX             { gde CX, tj. kraj linije. Ako je }
                PUSH Py             { brojač nula izađi iz rutine, u  }
                CALL GetPixel       { suprotnom ponovi ceo ciklus.    }
                POP  BX             { Subrutina @recur ispituje da li }
                POP  CX             { je piksel čija je pozicija odre }
                CMP  AX,BojaPoz     { đena preko BX i Py slobodan, što }
                JNE  @exit1         { znači jednak BojiPoz. Ako jeste }
                PUSH CX             { onda treba izvršiti rekurzivni  }
                INC  Brojac         { poziv, ali tako da se parametri }
                JMP  @start         { prosleđuju preko registara, a   }
  @backto:      POP  CX             { da se stare vrednosti skladišta }
  @exit1:       RET                 { na stek. Obratite pažnju da se  }
  @exit2:       NOP                 { BX ne stavlja na stek, jer je   }
end;                                { vrednost potrebna rutini niže ! }


var             GX, GM, Gaz, Vis, n1, n2, n3 : integer;

begin    {program}
    DetectGraph(GX,GM); InitGraph(GX,GM,'');
    Duz:=GetMaxX div 2; Vis:=GetMaxY div 2;
    for n1:=1 downto 0 do begin
        SetFillStyle(n1,15);
        for n2:=2 to 1 do begin
            for n3:=2 to 3 do begin
                Bar (Duz+(n3*50-n1), Vis+(n2*50-n1),
                     Duz+(n3*50+25+n1), Vis+(n2*50+25+n1));
                Bar (Duz+(n3*50-n1+20), Vis+(n2*50-n1+20),
                     Duz+(n3*50+45+n1), Vis+(n2*50+45+n1));
            end;
        end;
    end;
    Rectangle(0,0,GetMaxX,GetMaxY);
    AsmFill(1,1,0,Ofs(Uzorak));
    repeat until KeyPressed;
    CloseGraph;
end  {program}
```

Program za popunjavanje zatvorene konture uzorkom od 8x8 piksela

jednom bojom već uzorkom od 8×8 piksela te da je glavna rutina prirodno napisana u asemblerskom jeziku. Direktivom **$G+** definisana je upotrebe proširenog seta instrukcija 80286-ice. Rekurzija je ostvarena prenosom parametara putem registara, dok se važni podaci (iz BX i CX) privremeno čuvaju na steku.

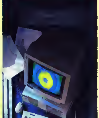
Specifičnosti algoritma omogućile su da se rekurzivni pozivi izvode sa JMP umesto sa CALL, čime se štedi na steku. Zaista, kada se porede obe rutine, R_FILL i ASM_FILL, jasno se vide prednosti integrisanog asemblera. Pored toga što sve teče žustrije, troši se i daleko manje steka: potrebno je manje od 3 K da bi se uzorkom ispunio ekran i prostor između iscrtanog lika. Rutina se uglavnom oslanja na standardne pascalove procedure **GetPixel**, **Line** i **PutPixel** koje doduše nisu brze, ali omogućuju kompatibilnost sa različitim grafičkim karticama. Sve ostale detalje možete naći u dopisanom komentaru.

476 [PJ.PASCAL; PC] Jeste li nekada poželeli da spisak naredbi za rad sa tekstualnim datotekama u *Turbo Pascal*-u dopunite (iz raznih razloga izostavljenom) funkcijom *seek*? Pogledajte sliku 1

Postavljanje pointera iza kraja datoteke ili njegovo „nišanjenje" na neki <CR> karakter izaziva očekivano ponašanje tj čitanje praznog stringa No, ako ga postavite na <LF>, biće učitan taj <LF> i čitav sledeći red Stavljanje nule u CX DX i broja 2 u AL omogućava očitavanje dužine tekst fajla što *Turbo Pascal* takođe normalno ne podržava [Preuzeto sa Sezama, autor **Božidar Veselinović** (*vbole*)]

477 [OS.WINDOWS; PC] *Windows 3 1* bi trebao da bude kompatibilan sa QEMM-om 6 02, ali uz pažljivu instalaciju Izvršite OPTIMIZE a zatim pokrenite programčić QWINFIX (isporučuje se uz QEMM) koji će u SYSTEM INI dopisati *SystemROMBreakPoint=false* Ovo je ekvivalentno pokretanju *Windows*-a opcijom /D S koja, prema dokumentaciji, zabranjuje *Windows*-u da konsti ROM adresni prostor između F000 i 10000 Ukoliko i dalje ima problema, *Windows* se može pokrenuti opcijom /D X, koja će isključiti kompletan segment A000-FFFF iz zone koju *Windows* pretražuje u potrezi za slobodnim prostorom Ovo je, inače ekvivalentno sa dodavanjem klauzule *EMMExclude=A000-FFFF* u SYSTEM INI [Preuzeto sa Sezama autor **Zoran Kehler** (*zkehler*)]

478 [OS.WINDOWS, PC] Malo zabave za korisnike *Windows 3 1* Aktivirajte *Program Manager* i kliknite na *Help/About* Zatim držeći pritisnut Shift i Ctrl dva puta kliknite na *Windows bmp* u gornjem levom uglu prozora Zatim kliknite na Ok



Ponovite čitavu operaciju i neko će vam mahati No to i dalje nije sve Ponovite istu operaciju još jednom (bez klika na Ok ovoga puta) i uživajte! [Preuzeto sa Interneta autor **Dave Krasnow**]

479 [OS MSDOS, PC] Često govorimo o raznim verzijama MS DOS-a, no znamo li baš za svaku od njih? Slika 2 je sistematičan pogled na taj problem [Preuzeto sa Sezama autor **Goran Ristić** (*gristic*)]

480 [PJ CLIPPER; PC] Kada se *Clipper*-ov sistem za komunikaciju sa diskom približi granici od 20 otvorenih datoteka mogu se očekivati čudni efekti u dibageru nestaje deo teksta, pojavljuju se poruke o nepostojećim greškama i slično Tada treba izvršiti DOS komandu SET CLIPPER = Fxx (gde je *xx* broj veći od 20) i sve će se srediti Alternativno program se može startovati sa PROG //Fxx [Preuzeto sa Sezama autor **Nenad Batoćanin** (*nbatocanin*)]

481 [SIS, PC] Jeste li nekada pokušavali da iz programa utvrdite da li je tastatura otključana ili zaključana? Mozda u okviru neke nove zaštite? Pogledajte sliku 3 program nije preporučljivo pustiti da radi u petlji dovoljno je jedno ispitivanje [Preuzeto sa Sezama autor **Božidar Veselinović** (*vbole*)]

482 [ZAST BASIC.452, PC, TIM] Sećate li se „trika" za razbijenje zaštite bejzik programa koji smo opisali pre dva meseca? U međuvremenu nam se javio čitalac koji predlaže još jednostavnije rešenje – učitajte zaštićeni bejzik program sa LOAD, a zatim ga snimite u ASCII obliku sa SAVE „IME" A i dobićete listing čitljiv u bilo kom editoru!

Probali smo i nikako nije išlo najzad se pokazalo da se tehnika može primeniti u CP/M

Listing 1

```
[illegible]
```

Funckija *seek* na *TurboPascal-u*

Slika 2

```
[illegible]
```

Sve verzije DOS-a

Listing 3

```
[illegible]
```

Ispitivanje statusa (otključana/zaključana) tastatura

Listing 4

```
[illegible]
```

Primer korišćenja određenih svojstava porta 3BAh

Slika 5

```
[illegible]
```

Tonsko biranje brojeva

bejziku na TIM-u 011, ali na i na PC-jevom GWBASIC-u o kome smo govorili. Ukratko i *Microsoft* je ponešto naučio između CP/M-a i MS DOS-a. [Prilog **Predrag Supurović**, Beograd]

483 [OTXT WPERF; PC] Mali (ili malo veći?) bag *WordPerfect*-a. U okviru *Format* (Shift F8) menija možete de zahtevate numerisanje redova (*Format* L N) koje će počinjati od **1** na svakoj strani ili se protezati kroz čitav dokument. Ma za koju se opciju opredelili, numerisanje redova će se vraćati na 1 na početku svake strana (?) U to ćete verovati dok posmatrate *Preview*, dok je na štampaču sve u redu. Čudna nedoslednost autora *WordPerfect*-a [Preuzeto sa Sezama, autor **Domen Ferbar** (*dferbar*)]

484 [TEHP; PC] Kada animirate neki lik, poželjno je da kretanje bude usklađeno sa periodom osvežavanja slike na monitoru: mala je korist od brze rutine ako je pokret izlomljen baš zbog ove neusklađenosti sa tzv. *vertical retrace*-om. Dovoljno je sačekati da se prikaže čitav *frame* i svaki će pokret izgledati savršeno glatko, zahvaljujući, pre svega, tromosti koja ne može da primeti kako kretanje nije kontinualno.

Na *Hercules* kartici ovaj problem se rešava stalnom kontrolom sedmog bita porta 3BAh – stanje 0 označava da je mlaz upravo stigao na dno ekrana i krenuo nazad u gornji levi ugao. To može da bude signal vašem programu da je trenutak pogodan za promenu, pomeranje sadržaja ekrana (ako radite *scroll*) i tome slično. Čim mlaz stigne na vrh ekrana, stanje ovog bita postaje jedinica. Nulti bit istog porta označava (za animaciju manje značajan) *horizontal retrace*, dok bit 3 sadrži vrednost video signala koji tog trenutka stiže na katodnu cev.

Na slici 4 prikazan je jednostavan primer korišćenja opisanih svojstava porta 3BAh [Preuzeto sa Sezama, autori **Aleksandar Vasić** (*vasic*) i **Dragan Zakić** (*dzakic*)]

485 [PJ CLIPPER, C, PC] Da bi se *Clipper* program povezao sa potprogramom pisanim u *Microsoft C*-u, C segment mora da se prevede uz korišćenje sledećih opcija:

/c – samo kompajliranje, bez pozivanja linkera

/AL – LARGE model: obavezno za sve rutine koje se linkuju sa *Clipper*-om, jer je i sam *Clipper* preveden u LARGE modelu

/Gs – bez generisanja koda koji proverava stek.

/Zl – bez stavljanja imena *default* biblioteka u objektni kod

/FPa – bez pozivanja rutina koje rade sa racionalnim brojevima

Treba, osim toga, znati da MSC stavlja *underscore* ispred imena funkcija u OBJ datoteci, što *Clipper* ne radi. Da bi se izbegle poruke linkera, treba sve funkcije u C segmentu deklarisati kao „pascal". *Clipper*-u neće smetati na ovaj način promenjeni način prenosa argumenata, jer se argumenti i tako prenose pomoću _par rutina.

Kada se linkuje sa MSC bibliotekama, mora se koristiti opcija /NOE da linker ne bi davao poruke o dvostruko definisanim simbolima – neki simboli su definisani i u *Clipper* i u MSC bibliotekama. Takođe je potrebno povećati broj segmenta na 512 ili više (/se 512). Da bi se najzad dobijeni kod mogao debagovati pomoću *Code View*-a, treba ga prevesti opcijom /Zi i generisati MAP fajl pri linkovanju (opcija /CO) [Preuzeto sa Sezama, autor **Zoran Kehler** (*zkehler*)]

486 [COMM] U poslednje vreme se dosta govori o sve popularnijem načinu biranja telefonskih brojeva koje se naziva tonsko ili tontrekventno. Ovaj metod biranja brojeva (*dual-tone multi-frequency*, DTMF) je nastao u *Bell*-ovim laboratorijama i u Sjedinjenim Državama se koristi još od sredine šezdesetih godina. Kodiranje brojeva se vrši pomoću dve grupe tonskih frekvencija: grupe viših frekvencija (kolone) i grupe nižih frekvencija (redovi). Pritiskom na jedan od tastera koji se nalaze na aparatu generišu se dva signala od kojih je jedan iz niže a drugi iz više grupe frekvencija. Na primer, pritisak na taster 1 (slika 5) izaziva oscilovanje dva oscilatora, jednog sa frekvencijom 679Hz i drugog sa frekvencijom 1209Hz. Svakom broju, dakle, odgovara jedna kombinacija frekvencija koja se u telefonskoj centrali posebnim kolima pretvara u traženi broj. Kod modema se tonovi nižih i viših frekvencija dobijaju pod softverskom kontrolom.

Prednosti tonskog (u odnosu na „klasično" pulsno) biranja su višestruke, a najvažnija je znatno kraće vreme uspostavljanja veze, naročito kod brojeva koji se sastoje od cifara 7, 8, 9 i 0. Osim toga, brzina otpreme cifarskih informacija ne zavisi od cifre koja se bira – sve se završi za oko 5 ms po cifri. „Okretanje" Sezamovog broja 011-646-899, na primer, iz unutrašnjosti traje oko 3 sekunde tonskim, ili 12 sekundi pulsnim biranjem.

Većina u poslednje vreme instaliranih centrala ima mogućnost tonskog biranja (METACONTA, AXE). Da bi se ta mogućnost realizovala, treba pozvati 977 (prijava kvarova) i saopštiti želju. Nakon desetak minuta bićete pozvani iz centrale i zamoljeni da isprtiskate sve tastere na telefonu (potreban je, naravno, telefon koji ima tonsko biranje) da bi se proverilo da li je aparat ispravan. Ako je sve u redu, mogućnost tonskog biranja se dobija u narednih par minuta, koliko je radniku centrale potrebno da otkuca dva znaka, tj. da dodeli „klasu" vašem telefonskom broju (kod nekih centrala se za ovo mora pisati i posebna molba). Samim dobijanjem mogućnosti tonskog ne gubi se mogućnost pulsnog biranja. Takođe, ako jedan od dvojnika dobije mogućnost tonskog biranja, automatski je dobija i drugi. Visina pretplate ne zavisi od načina biranja.

Metoda tonskog biranja sve se više koristi i kada treba kablom ili radio vezom preneti veći broj naredbi. Većina *Panasonic* telefonskih sekretarica i telefax uređaja, na primer, omogućava daljinsko komandovanje tako da se sa drugog telefona mogu preslušati pristigle poruke, promeniti standardni odgovor, premotati traka i tome slično. Zato neki telefonski aparati ili DTMF uređaji imaju jednu kolonu tastera više (označenu sa A, B, C i D). Sve ove opcije mogu se koristiti i tamo gde ne postoje tonske centrale.

Gradovi u Srbiji u kojima su instalisane centrale koje imaju mogućnost tonskog biranja su: Bor (brojevi koji počinju sa 3), Beograd (brojevi koji počinju se 1), Kraljevo, Kruševac, Lazarevac, Mladenovac, Niš (brojevi koji počinju sa 332), Pančevo, Ruma, Sremska Mitrovica i Šabac [Preuzeto sa Sezama, autor **Nebojša Icić** (*nicic*)]

497 [IGRE TETRIS, PC] Stari dobri Tetris krije još po neku tajnu. Svakako znate da je za „pamćenje" skora korišćena *integer* promenljiva koja posle 32767 pređe na -32868 i posle se polako „penje" prema nuli... naravno, samo ako ste pravi majstor za Tetris.

Do negativnog skora se, međutim, može doći i za trenutak – startujte Tetris i pritiskajte *Draw Next* (1 na numeričkoj tastaturi). Za trenutak vas eto u minusu! [Preuzeto sa Sezama, autor **Ivan Lazarević** (*ilazarevic*)]

Priloge za ovu rubriku šaljite na adresu „Računari (za „Bajtove lične prirode") Bulevar vojvode Mišića 17, Beograd ili preko Sezama u vidu lične poruke uredniku (*mail write dejanr*)

STA IMA NOVO

Tržište

EVROPSKA BANKA I COMPAQ U BUDIMPEŠTI

Evropska Banka za Rekonstrukciju i Razvoj (EBRD – European Bank for Reconstruction and Development) je 14. i 15. aprila održala svoj prvi godišnji sastanak u Budimpešti uz pomoć mreže personalnih računara firme „Compaq Computer Europe". Od osnivanja u aprilu prošle godine, EBRD je direktno investirala preko 621 Ekija (780 miliona USD) u 20 raznih projekata širom Istočne Evrope. Ovo inicijalno ulaganje je imalo za cilj promociju privatne inicijative u okviru lokalnih ekonomija.

U budimpeštanskom konferencijskom centru se okupilo 2.000 delegata čiji je pristup tekst-procesorima i prevodilačke zahteve podminilo 50 Compaq DeskPro 386s/20 računara i nekoliko Compaq laptop računara korišćenih zajedno sa svojim desktop postoljima. Na računarima su se vrteli tekst-procesori na mađarskom, ruskom, nemačkom, francuskom i engleskom, a mreža, koja je radila pod Novell NetWare 3.11, je obezbedila pristupne tačke na svakom spratu susednog Novotel Hotela za delegate koji su imali sopstvene laptop računare.

TRILATERALNI DESKTOP

Prve verzije paketa za zaštitu i rekonstrukciju fajlova bile su prilično nespretne i delovale odbojno svima izuzev onoj klasi korisnika koje Amerikanci nazivaju imenom „technical users" (tehnički orijentisani korisnici). Tri *utility* paketa koje je firma „Symantec" nedavno pustila u prodaju, *Norton Desktop for Windows, 2.0, Norton Desktop for DOS 1.0* i *Norton Utilities for Macintosh 1.0* na najbolji način demonstriraju koliko su stvari napredovale. Ovim potezom „Symantec" konsoliduje osobine tri potpuno različita paketa u jednu jedinu aplikaciju u tri verzije. *Norton Desktop for Windows* donosi zaštitu od virusa, *Norton Desktop for DOS* nudi fajl-transfer i *drag and-drop* mogućnosti, a *Norton Utilities for Macintosh* kombinuju najbolje iz starih NUM i SUM paketa na jednom mestu.

Norton Desktop for Windows 1.0 je ponudio korisnicima mnoge najbolje mogućnosti paketa *Norton Utilities, Norton Backup* i *Norton Commander*-a i dopunio mogućnosti *Windows File Finder*-a „drag-and-drop" kopiranjem i štampanjem, bekapom i setom *file-viewer*-a. Ipak, bio je suviše spor i imao mnogo bagova koji su činili instalaciju i rad veoma pipavim. Verzija *2.0*, po ceni od 179 USD (149 USD do sredine maja, 49 USD za *upgrade*) ne samo da rešava propuste ranije verzije, već, prema rečima predstavnika firme, proširuje ukupnu funkcionalnost. *Norton Desktop 2.0* kompletno zamenjuje *Program Manager* verzije *3.0* i *3.1* i dozvoljava kreiranje grupa i ikona aplikacija na desktopu. „Drag-and-drop" funkcije omogućavaju kopiranje/premeštanje fajlova u drugi drajv ili direktorijum, brisanje, back-up, ili automatsko startovanje aplikacije prevlačenjem odgovarajuće ikone fajla nad ikonu aplikacije. Prema „Symantec"-u, nova verzija radi šest puta brže i uključuje *Macro Recorder*, startovanje makroa u tačno zadato vreme i po određenom redosledu i detekciju virusa.

Norton Desktop for DOS je zamišljen kao naslednik *Norton Commander*-a (koji će se ipak prodavati još izvesno vreme) i po istoj ceni nudi i iste ili slične opcije kao i *Windows* verzija. Tekstualna verzija interfejsa za DOS se ponaša vrlo približno svom grafičkom rođaku, i čak omogućava slične „Drag-and-Drop" funkcije

SKORO ILI TAČNO 82

Kada je Sem Lojd prvi put postavio zadatak iz naslova u jednim američkim novinama, ponudivši pri tome 1000 dolara onome ko ga reši (tada je $1000 bilo više para nego što je danas!), iz čitave Amerike je pristiglo preko milion odgovora od koga su svega dva (međusobno identična) bila tačna. Lepo bi bilo znati da li bi danas odziv bio veći... da zadatak nije dovoljno poznat i obrađen u literaturi.

Podsetimo se, pre svega, postavke. Na raspolaganju imamo tačno sedam različitih cifra i osam tačaka, od kojih treba formirati nekoliko brojeva, tako da njihov zbir bude što približniji broju 82. Osim tih cifara i tačaka, ne smeju se koristiti nikakve druge cifre, znaci ili operatori, i moraju se iskoristiti svih sedam cifara i osam tačaka. Tačka može da se koristi na dva načina – kao decimalni separator ili kao oznaka periodičnih brojeva. Ukoliko se radi o periodu koji se sastoji od više cifara, tačka se stavlja iznad prve i poslednje.

Prvo pitanje na koje treba odgovoriti je da li se broj 81.9 računa kao 82 ili ne? Iako su mnogi primetili ovaj problem, samo se *Biljana Srdanov* dosetila da ga obradi na pravi način. Posmatrajmo, na primer, broj 0.9 a zatim napišimo 10*0.9 = 9.9 = 9+0.9 odakle se dobija da je 9*0.9 =9 odnosno da je 0.9 =1. Ukratko, u usvojenoj notaciji postoje dva načina da se predstavi broj 1 (a samim tim i broj 82): 1 ili .9, tj. svejedno je da li ćemo predstaviti broj 82 ili broj 81.9.

Pošto smo to pitanje apsolvirali, da vidimo u kakvom ćemo obliku tražiti sabirke. Pošto nigde nije rečeno koliko tih sabiraka treba da bude (postoji samo ograničenje na sedam cifara), svejedno je da li ćemo ceo i decimalan deo broja pisati zajedno ili odvojeno – uvek ćemo „potrošiti" jednak broj cifara i tačaka. Opredelili smo se da posebno pišemo ceo, a posebno decimalni deo. Drugo, da bismo „stigli" do 82, moramo da formiramo neki dvocifreni broj koji će biti „dovoljno blizu" 82 (suma sedam jednocifrenih brojeva uvek je manja od 82). Pošto formiramo taj dvocifreni broj, ostaje nam još pet cifri i osam tačaka. U jednom broju se može iskoristiti najviše dva puta više tačaka nego što ima cifara (na primer, u broju .5. smo iskoristili jednu cifru i dve tačke), što znači da od pet cifara i osam tačaka možemo konstruisati sledeće kombinacije:

a) četiri broja oblika .X. i jedan jednocifren broj
b) dva broja oblika .X., jedan broj oblika .X i jedan broj oblika .X.X.
c) dva broja oblika .X.X. i jedan broj oblika .X.
d) tri broja oblika .X. i jedan broj oblika .XX.
e) tri broja oblika .X. i dva broja oblika .X

Kombinacije b) i d) jedva da je potrebno komentarisati, jer se sabiranjem brojeva takvog oblika ne može dobiti ceo broj koji ćemo dodati izabranom dvocifrenom broju i tako „stići" do 82. Varijanta d), na primer, otpada jer bi kod broja oblika .XX. obe cifre morale biti jednake, a to je u suprotnosti sa pretpostavkom zadatka (broj .X. možemo da napišemo kao Y/9, a kada saberemo tri broja takvog oblika, do celog broja nam nedostaje još Z/3, Q/9 ili 0 – a ni jedan od ta tri broja ne možemo da napišemo u obliku .XX.).

Varijante a), c) i e) ne možemo eliminisati ovako jednostavnim rezonom pa svaku od njih treba ispitati. Ručno ispitivanje bi možda malo predugo trajalo pa ćemo posao poveriti računaru (možda je bio problem što u vreme Sema Lojda nije bilo računara?). U petlji za svaku od tri moguće kombinacije ispitujemo sva moguća rešenja i ispisujemo ona koja zadovoljavaju uslove zadataka. Program sa slike 1 pronalazi čak 32 „suštinski" različita rešenja. Njihovim kombinovanjem (npr. dodavanjem decimalnog dela dvocifrenom broju), taj broj se može znatno povećati. Kao „zvanično rešenje" odabraćemo kombinaciju

81+.2.+.7.4.+.0.3.

Toliko o zadatku, a sada o nagrađenima: prvu nagradu od 8000 dinara dobio je *Bratislav Veljković* iz Paraćina za najdetaljnije obrazloženo matematičko rešenje zadatka. Preostale nagrade smo izvukli: 6000 dinara dobija *Mihajlo Dojnov* iz Batajnice, 4000 dinara *Nenad Orlović* iz Novog Sada dok nagrada za rešenje dostavljeno modemom (6000 dinara) pripada *Bošku Koprivici* iz Sremčice. Pohvale su zaslužili *Biljana Srdanov, Predrag Cvetković, Jožef Kralica, Slavko Kukrika, Branko Obradović, Miloš Prvulović, Nina i Branislav Tomić* i *Saša Žunić*.

Slika 1

```
program broj_82;
{ Prema programu Slavka Kukrike }
var q,u,v,r,t,y,s,z: integer;
begin
  for q:=70 to 81 do
  for u:=70-q to 81-q do
  [illegible]
  begin
    [illegible]
    for u:=2 to 6 do
    for r:=0 to u-1 do
    for t:=1 to r-1 do
    for y:=0 to t-1 do
    [illegible]
    begin
      [illegible]
    end;
  end;
  for q:=70 to 80 do
  [illegible]
  begin
    [illegible]
    for r:=2 to 6 do
    for t:=1 to r-1 do
    for y:=0 to t-1 do
    [illegible]
    begin
      [illegible]
    end;
  end;
  for q:=80 to 81 do
  [illegible]
  begin
    for u:=1 to 6 do
    for v:=1 to u do
    for t:=0 to u-1 do
    for r:=0 to r-1 do
    [illegible]
    begin
      [illegible]
    end;
  end;
end.
```

Program pronalazi čak 32 „suštinski" različita rešenja zadatka opisanog u tekstu

(NE)STABILNI TROUGAO

Evo jedne pomalo neobične ali (verujemo) zanimljive pitalice. Dat je ravni kvadratni sto stranice A=2 metra i model jednakostraničnog trougla stranice B=20 centimetara, načinjen od tankog ali čvrstog kartona. Postavite trougao na sto tako da što manji deo njegove površine bude u kontaktu sa površinom stola, a da trougao ipak ne padne na pod. Ne dolazi, naravno, u obzir da se površina trougla postavi normalno na sto, tj. da trougao stoji na svojoj ivici – mora biti položen na sto, tj. delimično stajati na stolu a delimično „lebdeti" u vazduhu.

Rešenja Pitalice, zajedno sa programima koje ste napisali, pošaljite na uobičajenu adresu „Računari" (za Dejanove pitalice), Bulevar vojvode Mišića 17, Beograd tako da pristignu pre 1. juna 1992. Sva pisma sa korektnim rešenjima konkurišu za novčane nagrade od 16,000, 12,000 i 8.000 dinara, dok će kuponi (ili njihove fotokopije) na koje je upisan identifikacioni broj učestvovati u godišnjem takmičenju rešavača Pitalica. Identifikacioni broj dobijate tako što u prve tri kućice upišete poslednje tri cifre nečijeg broja telefona, u sledeće dve godinu vašeg rodjenja i na kraj dodate dve cifre po izboru. Obratite pažnju da sva rešenja koja šaljete u toku godine nose isti identifikacioni broj. Osim običnom, rešenje ove Pitalice možete da pošaljete i elektronskom poštom posredstvom Sezama – koristite menije REDAKCIJA, PITALICE, REŠENJE ili, iz komandnog moda, pošaljite ličnu poruku korisniku pitalice (*mail write pitalice*). Sva rešenja pristigla elektronskom poštom ravnopravno sudeluju u izvlačenju nagrada i konkurišu za **specijalnu nagradu** od 12,000 dinara koja će biti dodeljena najboljem rešenju koje je dostavljeno modemom!

75: Nad stolom je ____________ % površine trougla.

Ime i prezime ____________

Adresa ____________

Mesto ____________

Kompjuter ____________ Vreme ____________

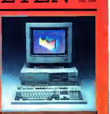
Identifikacioni broj: |_| |_| |_| |_| |_| |_| |_|
iz broja tel. god. rođ. po izboru

COMPAQ BILTEN

1
maj 1992

Priprema i uređuje: „MICROSYS"





Compaq DeskPro/M porodica personalnih računara

„COMPAQ" DESKPRO/M

"COMPAQ" je predstavio novu Deskpro/M familiju koju sačinjavaju šest modela u stonom („desktop") kućištu, sa EISA magistralom, projektovanih prema konceptu „inteligentne modularnosti". Za razliku od konvencionalnih PC računara kod kojih je najveći broj komponenti integrisan na matičnoj ploči novi „COMPAQ"-ov pristup je da sve komponente raspodeli u 5 različitih podsistema procesor EISA magistrala I/O video grafički kontroler i memorijsko proširenje, svaki na posebnoj ploči

Osnovni koncept

Koncept inteligentne modularnosti omogućava korisniku proširivanje ne samo procesora kako je to uobičajeno u većini PC sistema Korisnici mogu brzo i lako da modifikuju ključne sistemske komponente u skladu sa promenom svojih zahteva ili tehnološkim napretkom Pored toga održavanje ovakvih računara je znatno olakšano jer su glavne komponente na lako dostupnim modulima

Ova familija jasno odslikava „COMPAQ"-ovu želju i sposobnost za razvojem proizvoda koji predstavljaju solidnu osnovu za zadovoljavanje sadašnjih zahteva, definišući pri tome jasan put budućih proširenja „COMPAQ" do sada nije imao familiju proizvoda koja bi bila toliko univerzalna da se brzo lako i ekonomski efikasno može unaprediti praktično ceo sistem. Dakle korisnici mogu raditi razvoj na jednoj platformi, znajući da je njihova investicija zaštićena pošto se konfiguracije mogu lako izmeniti i prilagoditi novim potrebama i tehnologijama

Osnovni modeli

Šest novih članova ove familije su bazirani na „INTEL"-ovim procesorima „COMPAQ" DESKPRO 386/25M (procesor 386/25 MHz), „COMPAQ" DESKPRO 386/33M (procesor 386/33 MHz), „COMPAQ" DESKPRO 486s/16M (novi procesor 486SX/16 MHz), „COMPAQ" DESKPRO 486s/25/M (novi procesor 486SX/25 MHz), „COMPAQ" DESKPRO 486/33M (procesor 486/33 MHz), „COMPAQ' DESKPRO 486/50M (procesor 486DX2/50 MHz).

Članovi „COMPAQ" DESKPRO familije predstavljaju izvanrednu kombinaciju širokog spektra karakteristika i mogućnosti uz prihvatljive cene

Jednostavna nadgradnja

Pri projektovanju se vodilo računa o optimizaciji različitih karakteristika u cilju obezbeđenja funkcionalnosti pouzdanosti i racionalnosti Ovo uključuje specifično projektovan memorijski kontroler i druga ASIC kola koja se automatski prilagođavaju procesorima različitih brzina zatim otpornu čeličnu kutiju snažan izvor napajanja od 240 W koji je dovoljan za maksimalno konfigurisane današnje sisteme, kao i za buduće, na bazi novih procesora i ekspanzionih ploča

Program zamene procesorskih ploča ilustruje brigu za korisnika i zaštićenost njegovih investicija Kada korisnik odluči da procesorski modul zameni jačim (npr 386/33 za 486/33), vratiće stari i doplatiti samo razliku u ceni ova dva procesorska modula

Novi modularni dizajn omogućava da su izmene kratke i vrlo brze završavaju se za manje od 5 minuta Ovakav dizajn smanjuje sve zastoje koji se tipično odnose na modifikacije i održavanje. EISA magistrala omogućava priključivanje sadašnjih i budućih dodatnih modula Ova familija omogućava korišćenje svih 32-bitnih EISA modula a takođe je kompatibilna i sa kompletnim asortimanom od više hiljada postojećih 8- i 16-bitnih ISA modula

Standardne karakteristike i opcije

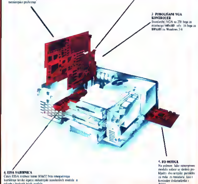
Svi modeli su standardno opremljeni sa 4 MB 80 ns

[illegible]

[illegible] tehničkih mogućnosti i performansi mašina svojevremeno nazvanih „personalni računari" [illegible] i značajno povećava Oblast njihove [illegible] je vrlo široka počev od „personalnih" [illegible] do profesionalnih koje zahtevima za računarskom snagom zahtevima za pouzdanošću i raspoloživošću prevazilaze uobičajene okvire personalnih i duboko zalaze u domen mini i supermini računara

Imajući ove činjenice na umu „MICROSYS", kao autorizovani zastupnik firme „COMPAQ" pokreće u „Računarima" redovni „COMPAQ" Bilten koji treba da posluži blagovremenom informisanju naše stručne javnosti o tehničkim novostima iz domena proizvodnje i primene „COMPAQ" proizvoda. Sadržaj „COMPAQ" Biltena će predstavljati stručni tekstovi o pojedinim [illegible] „benchmark"), primeri tehničkih rešenja („case studies" iskustva) naših i domaćih korisnika, kao i kratke vesti.

Vaš „MICROSYS"



32 bitnog RAM-a na procesorskoj ploči (sem modela DESKPRO 50M koji ima 8 MB) što se može proširiti do 64 MB dodavanjem posebnog memorijskog modula
Svi računari imaju 7 priključnica za proširenja 1 za procesor (širine 32 bita) 1 za memoriju (32 bita), 1 za video (32 bita) i 4 EISA priključnice širine 8/16/32 bita

Standardni VGA (izuzev DESKPRO 50 M) podsistem omogućava prikazivanje 256 boja pri rezoluciji 640×480 odn 16 boja pri rezoluciji 800×600 Ugrađen je grafički akcelerator koji poboljšava performanse video sistema za 50 % Model DESKPRO 50 M je standardno opremljen „COMPAQ" QVision 1024/E kontrolerom koji omogućava prikazivanje 256 boja pri rezoluciji 1024×768 odn 65536 boja pri rezoluciji 640×480 ili 16 7 miliona boja pri rezoluciji 640×400 bez preplitanja, sa učestanošću osvežavanja 72 Hz

Takođe su standardna dva serijska priključka Centronics, priključak za tastaturu i za miša I/O ploča sadrži kontrolere za jedinice disketa, diskove i traku. Svaki računar je opremljen po jednom disketnom jedinicom (3 5"/1 44 MB) a pored toga postoje još dva dostupna mesta 1/2 visine i jedno mesto 1/3 standardne visine, za jedinice masovnih memorija Standardni modeli su opremljeni diskovima kapaciteta 60 120 340 i 510 MB Maksimalno mogući kapacitet diskova u kućištu je 1 02 GB odn 18 2 GB sa dodatnim kućištem za proširenje pri čemu se mogu koristiti poboljšani ESDI ili 32-bitni IDA (*Intelligent Drive Array*) kontroleri Na raspolaganju su jedinice trake kapaciteta 60 do 525 MB kao i DAT (*Digital Audio Tape*) jedinica kapaciteta 1 3/2 0 GB

Izvor napajanja je predviđen za maksimalno opterećenje od 240 W sa višebrzinskim ventilatorom obezbeđuje napajanje za potpuno konfigurisan sistem sa sadašnjim i budućim procesorskim i ekspanzionim modulima

Obezbeđenje sistema od neovlašćenog korišćenja pored standardnih „COMPAQ"-ovih tehnika uključuje i dve nove „Asset Management Provision" za regulisanje prava pristupa informaciji o konfiguraciji i identifikaciju serijskog broja, pri radu u mrežama i „Quickblank" koji onemogućava da se podaci prikažu na ekranu Hardverske zaštite uključuju „EISA Configuration Lock" za zabranu neovlašćene modifikacije SETUP-a i „Cable Lock Provision" za fizičku zaštitu sistema od krađe

Novi član COMPAQ familije portabl računara COMPAQ je 27 04 1992 predstavio novog člana svoje familije portabl računara COMPAQ PORTABLE 486 Karakteristika novog modela je crno-beli TFT monitor sa aktivnom matricom standardne VGA rezolucije sa 64 nivoa sivog koji se odlikuje oštrom slikom visokim kontrastom i širokim uglom osmatranja Slično ranijem modelu 486c ovaj računar standardno sadrži 4 MB RAM-a (max 32 MB) jedinicu disketa 3 5", disk kapaciteta 120 MB (vreme pristupa < 19 msec), 2 EISA utičnice za proširenja CD-ROM audio i priključnicu za spoljašnji monitor u boji, kao i hardversku i softversku zaštitu („Compaq" Multilock koji uključuje 14 različitih hardverskih i softverskih alata za zaštitu) Dimenzije su 39.5 x 28 x 14 cm a masa oko 7 5 kg

Udruživanje vodećih proizvođača „COMPAQ" je nedavno objavio da je sklopio sporazume sa 15 različitih firmi, proizvođača hardvera, softvera i operativnih sistema Ideja je da se korisnicima obezbedi bolja podrška posebno u oblastima integracije sistema na bazi „COMPAQ" računara, gde učestvuje više proizvođača opreme Dakle, ako se problem pojavi treba ga rešiti, a ne prebacati na drugog isporučioca (prevazići „ping pong" sa korisnikom) Cilj je takođe i razvoj novih programa i proizvoda i zadovoljavanje tržišnih zahteva i mogućnosti posebno u domenu mreža i komunikacija Ovi sporazumi (tzv *Technical Support Alliances*) su sklopljeni sa sledećim firmama 3COM Corp „Banyan Systems Inc", „Digiboard Inc" „Digital Communications Associates", „Emerald Systems Inc", „Interactive Systems", „Liberty Electronics" „Microsoft Corp", „Novell, Inc" „Oracle Corp" „Proteon, Inc" „Racal Interlan Inc" „The Santa Cruz Operation (SCO)", SMC („Standard Microsystems Corp") i „Western Digital Corp"

QVision grafički sistem „COMPAQ" je nedavno predstavio novi grafički sistem, predviđen prvenstveno za grafičke aplikacije Osnovu sistema predstavlja video grafički kontroler visoke rezolucije QVision 1024 koji postoji u dve verzije 1024/E za EISA i 1024/I za ISA sabirnice Podržava monitore bez „preplitanja", sa učestanošću osvežavanja do 72 Hz tako da se dobija kvalitetna slika Maksimalna rezolucija je 1024×768 sa jednovremenim prikazivanjem 256 boja iz palete od 16 7 miliona Mogući režimi rada su i 800×600 sa 65536, odn 640×400 sa jednovremenim prikazivanjem 16 7 miliona boja Kontroler standardno ima 1 MB VRAM-a hardverski kursor i akceleratore za *Windows* i *AutoCAD* (tzv „line drawing machine") Softverski drajveri su optimizirani za *Windows AutoCAD* OS/2 PM i SCO UNIX X 11R4 Na raspolaganju su dva tipa ravnih „multi-scanning" kolor monitora visokih performansi QVision 150 (dijagonala 15") i QVision 170 (17") QVision je u potpunosti kompatibilan sa VGA i poboljšanim VGA standardima Prvi testovi brzine (na bazi kombinacije od 9 različitih grafičkih operacija) pokazuju da je za oko 10 puta brži od kontrolera na bazi TSENG ET 4000-E ili 2 5 puta brži od IBM XGA kontrolera

Sajamski vodič

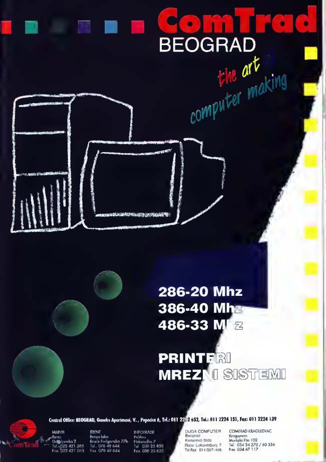
TEHNOTRONIKA '92.

36. međunarodni sajam tehnike Beograd, 11—16. 5. 1992.



U sajamskom vodiču vam se predstavljaju:


MB RAM, FDD 1,2/1,4,
1024×768 monitor,
baby kućište
1.950 DEM
368/33, 128 4 MB
RAM, FDD SuperVGA
1024×768 105 MB HD,
miš, tower DEM
TEHNOTRONIKA '92
Hala XIV, štand 1511



ComTrad
BEOGRAD
the art
computer making
286-20 Mhz
386-40 Mhz
486-33 Mhz
PRINTERI
MREZNI SISTEMI
Central Office: BEOGRAD, Geneks Apartmani, V., Popocica 6, Tel.: 011 22?2 652, Tel.: 011 2224 151, Fax: 011 2224 139
ComTrad
MANIR
Ruma
Gr?ljanska 2
Tel.: 022 421 265
Fax: 022 421 265
IDENT
Banja Luka
Braće Podgornika 77b
Tel.: 078 49 644
Fax 078 49 644
INFOTRADE
Priština
Hakundha 7
Tel 038 25 830
Fax 038 25 622
DUGA COMPUTER
Beograd
Kanarevo brdo
Raze Luksemburg 7
Tel/fax 011/587-468
COMTRAD KRAGUJEVAC
Kragujevac
Maršala Tita 102
Tel 034 34 270 / 60 336
Fax 034 67 117

MOST COMPUTERS

Beograd, Kralja Bodina 2, tel. 011 422 170, fax 011 426 144

COMPUTERS HOLLAND · COMPUTERS GERMANY · COMPUTERS YUGOSLAVIA

RAČUNARI • ŠTAMPAČI • SOFTVER • RAČUNARSKE MREŽE • DILERSKA PRODAJA

MOST JUNIOR
286/16MHz
1,490 DEM

MOST EXPERT
386/25MHz
2,090 DEM

MOST ASSISTANT
386/33MHz
2,390 DEM

MOST GENIUS
386/40MHz
2,490 DEM

Svi računari su opremljeni sa **IDE tvrdi disk "Microscience" 52MB sa internim cache-om, floppy 5 25" ili 3 5", "Supercom" monitor, "Hercules" grafički adapter, ASCII tastatura i uputstvo na srpskom jeziku. *286 i 386 računari:* 1 ili 2MB RAM-a, mini tower kućište. *486 računari:* "Soyo" osnovna ploča, ETEQ chipset, 4MB RAM-a i tower kućište.**

MOST GOLIATH
486/33MHz
3,590 DEM

DOPLATE ZA VEĆE ZAHTEVE

Stavka	Cena
Mono VGA monitor "Supercom", SVGA grafička karta "OAK" OTI-037, 800x600/256kB	150 DEM
Kolor VGA monitor "Huyndai", SVGA grafička karta "Trident" TVGA 9000A, 1024x768/512kB	590 DEM
Tvrdi disk 100-130MB	500 DEM
Tvrdi disk 200-210MB	900 DEM
Floppy disk 3.5"/1.44MB	159 DEM
RAM 1MB	125 DEM
Matematički koprocesor 387/12MHz	190 DEM
Matematički koprocesor 387/25MHz	380 DEM
Matematički koprocesor 387/33MHz	420 DEM

ŠTAMPAČI

Stavka	Cena
"EPSON" LX 850, 9-pinski, A4	690 DEM
"EPSON" LQ 570, 24-pinski, A4	790 DEM
"EPSON" LQ 1070, 24-pinski, A3	1,295 DEM
Laserski štampač "Hewlett Packard" HP LJ IIIp	3,450 DEM
Laserski štampač "Hewlett Packard" HP LJ III	4,800 DEM

Kupovinom računara »Most« dobijate sve što biste mogli da očekujete od računara firme sa velikim imenom, osim velike cene. Pored toga, naši računari pokriveni su garancijom od godinu dana, i obezbeđenim servisom nakon tog roka.

Tehnička podrška – svakog radnog dana od 10-14h. HOTLINE 011-672-682.

Ovlašćeni distributer – u Sava Centru, radnim danom od 9-17h. Tel. 011-222-43-22/492.

Potražite nas na Sajmu tehnike u Beogradu, na galeriji Hale XIV, od 11. do 15. maja '92



MOST design Boehnas Vujović

ERC POPUŠTA U SVEMU — OSIM U KVALITETU!

Specijalni sajamski popust do 20 maja 1992.



PC RAČUNARI (KOMPLETNE KONFIGURACIJE)

PC MODEL 286/16

Kućište Model 402 Flip Top
Headland Mainboard CPU 286/16 MHz sa 1 MB RAM
Multi I/0 karta sa 2 × seriel 1 × paralel 1 × game
Hercules karta
Floppy 1.2 MB
Hard disk 40 MB Western Digital ili Fujitsu
Tastatura 102 tasta Chicony YU
Monitor 14" Monochrom

Kompletna cena	DM 1520 –
VGA Mono	DM 1640 –
Color verzija	DM 2200 –
sa Color PANASONIC monitorom	DM 2300 –

MODEL 386/25

Kućište BABY 402D
Mainboard 386/25 4 MB RAM
Multi I/0 karta sa HDD FDD Kontroler
2 × seriel 2 × paralel 1 × game
VGA Karta 800 × 800 25 GKB RAM
Floppy TEAC 1.2 MB
HDD 40 MB Western Digital ili Fujitsu
Tastatura 102 tastera CHICONY YU
VGA mono monitor 14 800 × 800

Kompletna cena	DM 2395 –
Color VGA PANASONIC	DM 3050 –

MODEL 386/33

Mainboard 386/33MHz 64 Cache
Hard disk 40MB fujitsu
Ostala konfiguracija kao kod modela 386/25

Kompletna cena	DM 2550 –
Color VGA PANASONIC	DM 3200 –

MODEL 386/40 64 cache

Konfiguracija kao model 386/33
sa 40 MB Hard disk

Kompletna cena	DM 2700 –
Color VGA PANASONIC	DM 3350 –

MODEL 486/33 256 cache

Konfiguracija ista kao i kod 386/40
sa 40MB Hard diskom

Kompletna cena	DM 3110 –
Color VGA PANASONIC	DM 4260 –
DOPLATA ZA HARD DISK WESTERN DIGITAL 125MB 15ms	DM 420 –

MODEL 386 SX 16

Kućište BABY 402D
Mainboard 386 SX 16 1MB RAM
Multi I/O karta sa 2 × serijski
1 × paralalni 1 × game
VGA 800 × 800 256Kb
Floppy 1.2 MB
Hard disk FUJITSU 40MB
Tastatura 10% klik CHICONY YU
VGA Mono monitor 14"

Kompletna cena	DM 1820 –
Color VGA PANASONIC	DM 2500 –

MODEL 386 SX/25

Mainboard 386 SX 25MHz
Ostala konfiguracija ista kao i kod MODELA 386 SX 16

VGA Mono verzija	DM 1950 –
Color VGA PANASONIC	DM 2600 –

ISPORUKA ODMAH SA LAGERA
GARANCIJA 2–3 GODINE.
SERVIS OBEZBEĐEN U BEOGRADU
PLAĆANJE 100%

ERC commerc
SIRMIUM COMMERC
MEŠOVITO PREDUZEĆE EXPORT-IMPORT

Kolarčeva 4/IV
Beograd
Tel 011/632-532
011/626-792
Fax 011/633-059

HERMANN PUCHMANN
IM-EXPORT
HAMBURG Germany
STUDIOTECHNIK PUCHMANN D-2000 HAMBURG 70

ERC commerc

SIRMIUM COMMERC

Kolarčeva 4/IV
Beograd
Tel 011/632 532
011/626-792
Fax 011/633-059

HERMANN PUCHMANN
IM-EXPORT
HAMBURG Germany

MEŠOVITO PREDUZEĆE EXPORT-IMPORT

STUDIOTECHNIK PUCHMANN D-2000 HAMBURG 70

KONFIGURACIJA JE VAŠ – A KVALITET NAŠ POSAO!



Modeli 286, 386 i 486
u svim mogućim varijantama

Specijalno za vas softver
(za export-import firme,
agencije, zanatske radnje)
MS-DOS, Novell

Garancija 2 godine.

Isporuka sa lagera odmah
po uplati

Specijalni sajamski popust
do 20. maja 1992

Izaberite komponente sa spiska — dobićete sklopljen i konfigurisan računar.

KUĆIŠTA	DEM
MODEL 102 D BABY napajanja 200 W	155
MODEL 109 D BABY napajanje 200 W	180 –
MODEL 501 D MINI TOWER napajanje 200 W	180 –
MODEL 388 D MIDI TOWER napajanje 200 W	235 –
MODEL 703 D MIDI TOWER napajanje 200 W	220 –
MODEL 272 D SLIM LINE napajanje 200 W	205 –
MODEL 276 D SLIM LINE napajanje 200 W	220 –
MODEL 389 D BIG TOWER napajanje 230 W	330 –
MODEL 498 D BIG TOWER napajanje 230 W	330 –
MODEL 287 LAN STANICA napajanje 200 W	240 –
DISK JEDINICE	
MITSUMI 5 25" 1 2 MB	150
MITSUMI 3 5" 1 44 MB	150 –
TVRDI DISKOVI	
FUJITSU 40 MB 24 ms	495 –
WESTERN DIGITAL 125 MB 15 ms	910 –
CONER 215 MB 15 ms	1 650 –
TASTATURA	
CHICONY 102 TASTERA + YU	88 –
MATIČNE PLOČE	
286/16	190
386/SX 16	365 –
386/DX 25	680 –
386/DX 33 64 CACHE	830 –
386/DX 40 64 CACHE	980 –
486/DX 33 256 CACHE	690 –
TELEFONI	
MARKA COBRA BEŽIČNI domate 500 m	
MODEL 486 (crni, slonova kost, sivi)	175 –
MODEL 468 (beli)	230 –
MODEL 473 (beli)	350 –
MODEL 6525 sa telefonskom sekretaricom (beli)	370 –
Adapter 220 V na 9 V	19 –
Adaptar 220 V na 12 V	22 –

MEMORIJE	
SIMM SIPP 1MB 70 ns	26
44256 70	14 –
41256 70	3 50 –
MONITORI	
Monohromatski 14" Amber ili B/W	225
VGA Mono 14" 800×600 B/W	280 –
SVGA Color 14" 1024×768	800 –
SVGA Color 14" PANASONIC	900 –
SVGA Color 14" PHILIPS	900 –
GRAFIČKE KARTICE	
HERCULES	35 –
EPROM SA YU SETOM ZA HERCULES	15 –
VGA 800×600 256 Kb	95 –
SVGA 1024×768 512 Kb	150 –
ŠTAMPAČI	
STAR SC 24 15 A3 FORMAT 24 iglični	1 320 –
NEC P 30 A3 FORMAT 24 iglični	1 350 –
DISKETE	
FUJI MD 2 HD 1 2 MB	2 –
FUJI MD 2 HD 1 44 MB	3 60 –
MODEM	
2400 DISCOVERY extarni	480 –
2400 DISCOVERY interni	240 –
RAZNO	
Kontrolar AT BUS sa 2 × ser /1 × par port	52 –
MIS GENIUS GM 6	55 –
MIS GENIUS GM F303 SA PODLOGOM	80 –
EPROM PROGRAMATOR SUNSHINE ZA 4 EPROMA	390 –
ORIGINAL MS DOS 5.00 Microsoft	250 –
ORIGINAL MS DOS 4.01 (RUSKI) Microsoft	250 –
Nosač za HARD DISK 1 PAR (adapter)	6 –
Adaptar za flopi 3.5" na 5.25"	20 –
Centroniks kabl	18 –

VELIKI PLANOVI ... MALI BUDŽET ??

TEHNICOM

RAZGOVARAJTE SA INŽENJERIMA

Pre, posle i za vreme
SAJMA TEHNIKE!
Bul. Vojvode Mišića 37
11000 Beograd
Tel/fax: 011. 235.1108

- PC 286/386/486
- Štampači EPSON
- i Hewlett-Packard
- PANASONIC FAXovi
- MODEM

- Knjigovodstveni softver
- Škola i obuke
- Servis
- NOVELL i UNIX

ZOVITE NAS DA VAM POMOGNEMO DA IZABERETE VAŠE ALATE

1) Turbo Pascal - naredbe i objekti

Nasleđivanje, virtuelni metodi, klase, objekti, opis svih naredbi, prevod svih grešaka, ilustracija DOS i BIOS poziva, opis direktiva prevodiocu. Svi primeri napisani su korišćenjem objekata! **Svet Kompjutera, januar 91:** "Po preciznosti i sveobuhvatnosti izlaganja, ova knjiga zamenjuje odgovarajuće delove originalne dokumentacije." **Računari, februar 1991:** "Pouzdan i iscrpan priručnik u svakodnevnom radu..." 240 strana, autor: Duško Savić.

2) Turbo Pascal 6.0 - turbovizija i grafika

Opis svih metoda, kompletno objašnjenje svih klasa i cele hijerahije turbovizije. Programski primeri, editori, meniji, resursi, tokovi, dizajn interaktivne aplikacije... Izgradnja nove hijerarhije grafičkih objekata: mnogouglovi, matrice, linearne transformacije u 2D i 3D, perspektivna transformacija... Hiljade i hiljade naredbi! **Na ovo ste čekali!!!** 200 strana, autor: Duško Savić.

Knjige pod 1) i 2) su komplet za Turbo Pascal 6.0!

3) Uvod u objektno programiranje

Svet Kompjutera mart 92: "Primer studiozno organizovane i napisane knjige... Poglavlja su posvećena objektno orijentisanim programskim jezicima. Turbo Pascal 5.5, Turbo Pascal 6.0, TopSpeed Modula 2, C++ i Smalltalk/V... **Ako želite objektno da programirate neka ovo bude vaša prva knjiga iz te oblasti.**" 245 strana, autor: Duško Savić.

4) Efikasne baze podataka

Prvi deo: dBASE i efikasne baze podataka **za programera.** Drugi deo: JPI Modula 2 i efikasne baze podataka **za korisnika** (Programi iz drugog dela izvršavaju se 100-150 puta brže od onih u prvom.) **Za sve jezike i programere:** stranična organizacija podataka nasuprot indeksnoj, kako upisati datum u dva bajta umesto u osam... **Profesionalci!** Pišite *efikasne baze podataka*, i mušterije umesto da kupuju brži hardver neka više plate *Vas*. Knjiga za svakog dBASE i Clipper programera! 200 strana, autor: mr Dragoljub Savić.

Mogu se dobiti i diskete sa programima iz knjiga 1), 2) i 4)!

Cena po knjizi je dinarski ekvivalent od 10 DEM na dan uplate. Cena diskete po knjizi: 10 DEM. Isporuka pouzećem!

PC Program, Generala Hanrisa 20, 11040 Beograd, tel/fax (011) 463 296 žiro račun: 60816-603-36098

I dalje aktuelno: programi KUFI i KUIM - po 5000 din!

PC Art

PROJEKTUJE SOFTWARE

i u tome ima mnogo uspeha zato što zapošljava samo mlade i sposobne stručnjake

Ako želite da nam se pridružite a znate da servisirate PC računare i poznajete NOVELL operativni sistem javite se

Tel. 627 817, 620-647

agencija IN

POSAO PO MERI VAŠEG USPEHA

S Y B E X

ADVANCED TECHNIQUES IN AUTOCAD
SECOND EDITION
Robert M. Thomas

Knjiga najvećeg svetskog izdavača kompjuterske literature u svetu SYBEX a sada na srpskom jeziku

- verzija 10
- treća dimenzija i AutoLIPS
- pravljenje sopstvenih share font fajlova i modifikacija menija

Neophodno za arhitekte inženjere crtače.
Knjiga izlazi iz štampe u junu 1992. godine
Obim 464 strane
Informacije kod izdavača

B I P I F

MP za poslovne informacije i izdavačku delatnost d.o.o.

na tel. 327-891 fax 327 321

NARUDŽBENICA

Neopozivo naručujemo ____ primeraka knjiga AutoCAD — Napredne tehnike po ceni od 10 000 00 dinara (cena do 31.5.1992. godine). Navedeni iznos ćemo platiti po prijemu predračuna MP „BIPIF" d.o.o. Beograd Ul. 27. marta 38/I

Naručilac ____________ Mesto ________

Adresa ____________ Telefon ________

M.P. Potpis kupca

KOMPJUTER SERVIS
- SPEKTRUM
- COMODORE 64 i periferije
- AMIGA 500
- PC

Avalska 8 (na Slaviji)
tel. 011/434-287

YU SLOVA

LATINICA , ĆIRILICA ZA SVE *ŠTAMPAČE*
011/347-509, 403-229

Za sve informacije o cenama oglasnog prostora u „Računarima", rokovima predaje materijala, pogodnostima obratite se na

Tel. 011/653 748
Fax 011/648-140
Modem 011/648-899
(osoba za kontakt Esad Jakupović)

IBM PC & COMPATIBLE

...ponuda programa za PC ... iz svih oblasti

NOVO – BESPLATNO Najveći ... software-a (Public Domain) u zemlji ... **SVE! KATALOG** Besplatan savjet ... specijaliste EE Software F. G. Marica 31 78000 Banja Luka tel. 078/40 340. Nazovite naš BBS 078-40 940 svaki dan od 22h do 07h

PC AT 486-50 EISA

SC 4050 EISA

OSNOVNA PLOČA MICRONICS - Made in USA, CACHE 256 Kb, PROCESOR INTEL 486-50, RAM 8 Mb / Max 64 Mb NA PLOČI, EISA SCSI KONTROLER 4 Mb CACHE, HARD DISK 210 Mb MAXTOR, CONNER S3-SVGA CARD 1 Mb (*Šest puta brža od TSENG LABS ET-4000*), **14 " COLOR MONITOR 1024 x 768, FD 1.2 Mb + FD 1.44 Mb, TASTATURA 101, TOWER KUĆIŠTE**

9.950 DEM

HARD DISK 210 Mb, 15 ms .... 1 500 DEM
HARD DISK 360 Mb, 15 ms .... 2 400 DEM
HARD DISK 540 Mb, 15 ms .... 3 900 DEM
S3-SVGA CARD 1 Mb .... 550 DEM

	1 - 10 kom.	11 - 30 kom	preko 30 kom
TOWER KUĆIŠTE + 200 W	250 DEM	225 DEM	199 DEM
AT BABY CASE + 200 W	160 DEM	155 DEM	135 DEM
HERCULES MONO MONITOR 14 "	250 DEM	220 DEM	190 DEM
SVGA MONO MONITOR 14 " 1024 X 766	300 DEM	250 DEM	210 DEM
SVGA COLOR MONITOR 14 " 1024 x 766	750 DEM	690 DEM	595 DEM
EPSON PRINTER LX 400 A4 / 9 PIN / YU SET	540 DEM	500 DEM	470 DEM
EPSON PRINTER LQ 570 A4 / 24 PIN	790 DEM	750 DEM	690 DEM
EPSON PRINTER LQ 1070 A3 / 24 PIN	1150 DEM	1060 DEM	990 DEM
EPSON PRINTER FX 1050 A3 / 9 PIN / YU SET	1350 DEM	1260 DEM	1170 DEM
EPSON PRINTER DFX 5000 A3 / 9 PIN / heavy duty / 535 z/s	POZOVITE		

POLO
Tel.: 081/13-202
Fax: 081/12 684
PODGORICA

DISTRIBUTERI

EMI
Tel./Fax: 081/34-425
PODGORICA

UPLATA JE PO DNEVNOM KURSU DEM NA TRŽIŠTU U BEOGRADU
RADNIM DANOM od 9 do 17 h
ŽIRO RAČUN: 60815-601-64787

100 % AVANS
ISPORUKA ODMAH

GARANCIJA 12 MESECI - SERVIS I ODRŽAVANJE

TEL: 011/332-607
FAX: 011/345-126

BEOGRAD, 15.05.92.

PREDUZEĆE
ŠUTLIĆ
&
MIKRO DIZAJN

11000 BEOGRAD
Kosovska 32/I sprat

INSTITUT ZA NUKLEARNE NAUKE "VINČA"
CENTAR ZA PERMANENTNO OBRAZOVANJE
11000 Beograd, Nemanjina 4/X
Telefoni: 011/ 683-390, 682-486, 641-155/ 107,181
Telefax: 011/ 682-486

KNJIGE U OBLASTI PROGRAMIRANJA, PROGRAMSKIH JEZIKA I PRIMENE RAČUNARA

1. AutoCAD (verzija 11.0)
konstruisanje i projektovanje pomoću personalnih računara
Autori Boris Damjanović i Petar Damjanović
Sedmo izdanje, izlazi iz štampe u junu 92.

2. Uvod u C jezik
Autor Vladan Vujičić
Četvrto izdanje 1991 latinica 317 strana, format B 5 broširan povez

3. Primena programa SYMPHONY na personalnim računarima
Autor Dragan Pantić
Treće izdanje, 1990 latinica 226 strana format B 5 broširan povez

4. OS/2 - vodič za korisnike
Autor Zorica Jelić
Prvo izdanje 1989 latinica 253 strane format B 5 broširan povez

5. VENTURA - računarsko izdavaštvo
Autor Predrag Davidović
Treće izdanje izlazi iz štampe u junu 92

6. FORTRAN 77
standard sa dopunama za personalne računare
Autori Vlajko Kocić i Zoran Konstantinović
Drugo izdanje 1990 latinica 422 strane format B 5 broširan povez

7. UNIX - vodič za korisnike
Autor Zorica Jelić
Treće izdanje 1991 latinica 422 strane, format B 5 broširan povez

8. Primena programa FRAMEWORK III na personalnim računarima
Autor Dragan Pantić
Drugo izdanje 1991 latinica 326 strana format B 5 broširan povez

9. Programski alati u matematici
MathCAD, GRAPHER, EUREKA
Autor Ante Čurin
Prvo izdanje 1990 latinica 402 strane format B 5 broširan povez

10. Primena programa QUATTRO na personalnim računarima
Autor Dragan Pantić
Prvo izdanje 1992. latinica, 296 strana format B 5 broširan povez

11. DOS UKRATKO - verzije 3 3 i 5.0
Autor Dragan Pantić
Drugo izdanje 1992 latinica, 120 strana format B 5 broširan povez

12 Vodič za VAX/VMS
Autori Tamaš Kerepeš, Zvonko Oršolić i Saša Matijević
Prvo izdanje 1990 latinica 512 strana format B 5 broširan povez

13. Primena programa EXCEL na personalnim računarima
Autor Dragan Pantić
Prvo izdanje 1991 latinica, 272 strane format B 5 broširan povez

14. UNIX - vodič za programere
Autor Zorica Jelić
Prvo izdanje 1991 latinica 326 strana format B 5 broširan povez

15. WINDOWS 3 0
Autor Dragan Pantić
Prvo izdanje 1991 latinica 273 strane format B-5 broširan povez

16. PRIMAVERA
upravljanje projektima uz pomoć računara
Autori Jaroslav Urošević i Jelica Draškić- Ostojić
Prvo izdanje, 1991 latinica 365 strana format B 5 broširan povez

17. dBASE III + priručnik
Autor Milorad Filipović
Prvo izdanje 1991 latinica 249 strana format B 5 broširan povez

18. Osnovi informaciologije i informacione tehnologije
Autor Ljubomir Dulović
Prvo izdanje 1991 latinica 538 strana format B 5 broširan povez

19. LOTUS 1-2-3 (verzije 3.0 i 3.1)
Autor Dragan Pantić
Prvo izdanje 1991 latinica 411 strana format B 5 broširan povez

20. dBASE IV priručnik
Autor Ljubomir Lazić
Prvo izdanje 1991 latinica 306 strana format B 5 broširan povez

21. WORDPERFECT (verzija 5.1)
Autor Dragan Pantić
Prvo izdanje 1991 latinica 300 strana format B 5 broširan povez

22. Programiranje u CLIPPER-u 5.01
Autor Aleksije Veljović
Prvo izdanje 1992. latinica, 465 strana format B 5 broširan povez

23 FoxPro
Autor Dušan Čašić
Prvo izdanje 1992 latinica 390 strana format B 5 broširan povez

24. Uvod u strukture podataka
Autor Miroslav Jocković
Prvo izdanje 1992 latinica 357 strana format B 5 broširan povez

25. ORACLE (verzija 5.)
arhitektura i administracija
Autor Vladimir Milojković
Prvo izdanje, 1992. latinica 151 strana format B 5 broširan povez

"RAČUNARI", april 1992

Poručujem (ispod rednog broja knjige upisati broj poručenih primeraka)

Redni broj knjige	1.	2.	3.	4.	5.	6.	7.	8.	9.	10	11.	12.	13.	14.	15.	16.	17.	18.	19.	20.	21	22.	23	24.	25.
Broj poručenih primeraka																									

IME I PREZIME
(Naziv preduzeća) ____________________
ULICA I BROJ ____________________
BROJ POŠTE I MESTO ____________________ TELEFON ____________

Porudžbenicu poslati sa petim primerkom uplatnice (informacije o cenama svakog radnog dana od 8 do 20 sati) na adresu
Institut za nuklearne nauke "Vinča"
Centar za permanentno obrazovanje, Beograd, Nemanjina 4/X
Uplata se vrši na žiro račun
INSTITUT ZA HEMIJSKU DINAMIKU I PERMANENTNO OBRAZOVANJE, broj: 60803-603-17361.
Troškove poreza na promet i isporuke knjiga snosi poručilac i uplaćuje ih unapred zajedno sa uplatom za knjige.

MB SOFT
NOVI SAD

Program za personalne računare koji će te koristiti svakodnevno!

Standardni englesko-srpski srpsko-engleski

REČNIK ZA PC RAČUNARE verzija 2.0

- *PREKO 90.000 REČI IZ SVIH OBLASTI*
- *ELEKTRONSKI PREVODILAC*
- *GRAMATIKA*



- TRENUTNO nalaženje reči na srpskom ili engleskom jeziku
- mogućnost NEOGRANIČENOG dopunjavanja novim rečima
- "SOFT SEEK" - inteligentna pretraga
- učitavanje engleskog ili srpskog teksta u ASCII formatu i njihovo prevodjenje
- GRAMATIKA engleskog jezika
- "ON LINE" help
- VELIČINA PROGRAMA 3.8 MB
- 5 disketa i priručnik

Porudžbine ☎ (021) 612-752 i 368-476

ŠMIT ELECTRONIC

Kompjuterski program za poslovne ljude

MANAGER

- **ADRESAR** preko 100 firmi iz Srbije koje se bave prodajom hardvera softvera
- **PODSETNIK**
- **SELEŽNICA**
- **CENOVNICI** računara i računarskih komponenti nekoliko desetina firmi
- **KARAKTERISTIKE HARD DISKOVA** - preko 500 modela
- **KARAKTERISTIKE ŠTAMPAČA** - preko 100 modela
- **EVIDENCIJA** prometa profita i reshoda
  - + kalkulator, kalendar, časovnik
  - + help, štampano uputstvo

CENA 100 DEM

Posetite nas na sajmu TEHNOTRONIKA 92 štand **INTERSOFT & ŠMIT ELECTRONIC**

STARA PAZOVA, J. Čmelika 22, Tel. 022 31 18 09

design CD

BESPLATNI PROGRAMI!

izaberite između 7000 naslova

tel/fax. 011-647-190

ADD SYSTEMS
Jovana Ristića 6
Beograd

KUPUJTE PROVERENE DISKETE PUNE BESPLATNIH PROGRAMA!

Svaka disketa sadrži najmanje 700Kb kompresovanih programa!

Kompleti disketa za Clipper, TurboPascal, C, BASIC, Windows 3.0, AutoCAD, Fontovi, Sistemski programi, Laserski štampači itd. SHAREWARE & PUBLIC DOMAIN

Besplatan katalog sa preko 7000 naslova
Isporuka pouzećem ili lično
Posebne pogodnosti za preduzeća

ODKUP POLOVNE RAČUNARSKE OPREME!
PRODAJA RAČUNARA STARO ZA NOVO!
PRODAJA REPARIRANIH RAČUNARA!

ELHARD

25238 KULA
V. VLAHOVIĆA 46
Tel 025/722-740
Fax 025/723-433

Artikal	Cena
KUTIJA ZA DISK (100 kom, sa bravom) 5 25"	25 5
KUTIJA ZA DISK DD40L 3 5"	17
BRACKET FDD W/CABLE&ADAPTOR	10 5
BRAKET FOR HDD 3 5	3 5
C7** CO-PROCESSOR&CPU**	
C80287-12 IIT	170
C80387 16 SX INTEL	382 5
C80387-25	595
C80387-33	595
CABLE PC TO PRINTER	15.75
CABLE POWER PC EURO TYPE	6 75
CABLE RS232 TO RS232 F-M	17 5
CABLE RS232 TO RS232 M-M	17.5
CARD AT MULTI I/O 1P/1S/1G	20 48
CARD AT MULTI I/O 1P/2S/1C	29 75
CARD DISK MULTI I/O STD XT	52 5
CARD DISK MULTI I/O SUPER 1 44	68 25
CARD EGA 8 BITS	113 8
CARD ETHERNET 8BITS	245
CARD ETHERNET 16BITS	297 5
CARD HDD XT MEM	105
CARD HDD/FDD 1 2/2S/1G IDE	56
CARD HDD/FDD DTC 3280 SCSI	411 3
CARD HDD/FDD DTC 6280 ESDI	376 3
CARD HDD/FDD IDE 1 1	31 5
CARD HDD/FDD MFM 1.2	126
CARD MGP (HERCULES)	29 75
CARD VGA 16BIT 256K ONLY C&T	105
CARD VGA 1MB TSENG-LAB	245
CARD VGA 512K EXP 1M TRIDEN	161
KUĆIŠTE MINI TOWER WITH 200W	262 5
KUĆIŠTE C2KL WITH 200W	218 8
KUĆIŠTE C2K WITH 200W	210
KUĆIŠTE TOWER L WITH 220W	358 8
ČISTAČ GLAVE ZA FLOPI 3 5"	4 55
ČISTAČ GLAVE ZA FLOPI 5 25	4 55
DATASWITCH BOX 3 WAYS	35
DATASWITCH BOX 4 WAYS	38 5

DISKETE

Artikal	Cena
BLANK 3 5" DSDD	21
BLANK 3 5" DSHD	36 5
BLANK 5 25 DSDD	10 5
BLANK 5 25 DSHD	15 75
MAXELL 1 2MB	25 38
MAXELL 1 44MB	57 75
MAXELL 360KB	17 5

FLOPI DISKOVI

Artikal	Cena
FDD MITSUBISHI 3 5" 1 44	138 3
FDD MITSUBISHI 3 5" W/B	140
FDD MITSUBISHI 5 25" 1.2	148 8
FDD MITSUBISHI 5 25" 360	113 8
FDD SONY 3 5" 1 44	138 3
FDD PANASONIC 1 2MB	140
FILTER MONITOR 14" NYLON	26 25
FILTER MONITOR 14" ANTISTATIC	35
FILTER MONITOR 14" STAKLENI	49

HARD DISKOVI

Artikal	Cena
HD ST 351AX 42MB 24ms	472 5
HD3096A 3 5"80MB/ IDE	700
HD1144A 3 5"125MB/ IDE	875
HD 3 5"200MB/ IDE	1610
HD ESDI 320MB FH	3063
HD ESDI 660MB FH	3833
HD SCSI 320MB FH	3063
HD SCSI 1GB FH	5723
HD******* MAXTOR *******	
HD 7040A 40MB	500 5
HD 7080A 80MB	787 5
HD 7120A 130MB IDE	945
HD LXT-213A 200MB	1768
HD LXT 213S 200MB	1785
HD 340A 340MB	2835
HD 340S 340MB	2870
HD XT-4380S(333 4MB/16ms/SCSI)	3010

RAM

Artikal	Cena
I C 1MB 08 X 9 SIMM -08	119
I C 2MB 2*1MB SIMM -08	245
I C 256KB 10X9 SIMM-08	35
I C 41000 08	12 25
I C 41256 08	4 375
I C 44256 07	14
K BOARD 101 KEYS MSW ENG	66 5
K BOARD 101 KEYS JSW ENG	87 5

OSNOVNE PLOČE

Artikal	Cena
MBD 286 16G (10/22MHz)	201 3
MBD 286 20MHz	236 33
MBD 386SX-16 (8/16MHz)	437 5
MBD 386SX-20	490
MBD 386DX 25B (10/31MHz)	0
MBD 386DX 33C64B 55MHz)	787 5
MBD 386DX 40 C64K	875
MBD 486 33 C64K ORAM ETEQ	1943
MBD 486DX 50 C64K ORAM UMC	2800

MODEMI

Artikal	Cena
E 1800 INT 2400 Baud	210
E-1896 INT 2400/9600 Baud Send/Receive Fax	350
MOUSE FANCY F 360	68 5
MOUSE FANCY A5 F-840	56
MOUSE WITTY C 300	38 5

MONITORI

Artikal	Cena
COMPAL MUL 0 28/1024X768A	1085
CTX VGA 0 28/1024X768A	866 3
COMPAL VGA MONO	358 8
YANJEN VGA 0 39 1024X768	714
COMPAL 14" MONO	255 5
20" VGA 0 28/1024X768	3115

ŠTAMPAČI

Artikal	Cena
HP LASER IIIP (1MB 4PPM, 36 FONTS)	3129
HP LASER III (1MB 8PPM 36 FONTS)	5128
HP SCANJET 9195 W/CONTROLLER	3675
STAR NK 1000 II	547 8
OKI 183 A3/9PIN/120CPS	638 8
EPSON FX 1050 A3	1302
FUJITSU DL 3400	1356
TRAKA ZA ŠTAMPAČ STAR NX 1000 II	14
TRAKA ZA ŠTAMPAČ OKI ML 183	14
SCANNER WITTY MS-8000 400DPI OCR	264 3
TAPE BACK UP DI 10 120MB INT	855 8
TAPE BACK UP DI 20 250 MB INT	1138
TONER ZA HP II P	192 5
TONER ZA HP III	236 3
UPS 500W	752 5
UPS 1KW	1418



KAŽU DA JE PAS ČOVEKOV
NAJBOLJI PRIJATELJ
ALI...
POSTOJE SITUACIJE U KOJIMA
VAM PAS
NE MOŽE
POMOĆI !
ZMAJ MEĐU KOMPJUTERIMA
OTC
COMPUTERS
MI NE PRODAJEMO PSE. MI PRODAJEMO ZMAJEVE!
OTC JE PRVA TAJVANSKO-JUGOSLOVENSKA KOMPANIJA KOJA SE BAVI PROIZVODNJOM I PRODAJOM PC RAČUNARA.ŽELIMO DA VAM PONUDIMO RAČUNARE KOJI VAM MOGU POMOĆI U OSTVARENJU VAŠIH IDEJA I KREIRANJU NOVIH VREMENA
POZOVITE NAS
RADO ĆEMO VAM PRUŽITI DODATNE INFORMACIJE
MILENTIJA POPOVIĆA 9 - SAVA CENTAR - 11070 NOVI BEOGRAD - YUGOSLAVIA
TEL 38 11 222 23 55 TELEX 12042 YU SAVCEN - FAX 38-11-222 23 55 TEL 38-11-13 98 40 - EXT 684



INSTANT RADNA STANICA

JEDNOSTAVNO DODAJTE OPEN DESKTOP

Ako pogledate na veliku većinu grafičkih stanica koje su razvijene u toku prethodne decenije, primetićete nešto što sve imaju zajedničko

Integrisano UNIX okruženje.

Ako sada pogledate na veliku većinu firmi koje su u toku prethodne dekade moć računara stavile direktno u svoje desktop sisteme, videćete nešto što sve one imaju zajedničko

Personalne računare prema industrijskim standardima

Nije vam potreban računar da biste prognozirali platformu na kojoj će se grafičke stanice u ogromnoj većini firmi staviti u poslovne i inženjerske desktopove u **sledećoj** deceniji:

Integrisano UNIX okruženje za personalne računare prema industrijskim standardima

U tome je suština Open Desktop sistema

Open Desktop je kompletan grafički operativni sistem koji je izrađen na najpopularnijoj UNIX platformi svih vremena — SCO. Ona vam omogućava da uz pomoć ikona stvorite sopstvenu mrežu radnih stanica korišćenjem standardnih 386 ili 486 i periferala po vašem izboru

U jednom jedinom potpuno integrisanom paketu, koji se lako koristi i koji ima potpunu podršku, Open Desktop vam daje

* Punu 32-bitnu multitasking računarsku moć SCO UNIX sistema V/386
* Pridržavanje POSIX i X/OPEN standarda
* Grafički interfejs baziran na OSF/Motif i kompatibilan sa Presentation Manager-om
* Usluge upravljanja distribuiranom SQL bazom podataka
* Kompatibilnost sa postojećim DOS, XENIX i UNIX System aplikacijama i datotekama
* NFS, TCP/IP i LAN Manager mogućnosti umrežavanja

I sve to po neverovatno povoljnim cenama

Otkrijte kompletan grafički operativni sistem koji vodeće svetske kompanije biraju kao svoju platformu za razvoj za 1990-te i koriste da danas svoje PC računare 386 i 486 pretvore u instant radne stanice

Open desktop od SCO

OPEN DESKTOP
The Complete Graphical Operating System

ENERGOPROJEKT
ENERGODATA

SCO
THE SANTA CRUZ OPERATION

Generalni distributer za Jugoslaviju

ENERGOPROJEKT • HOLDING KORPORACIJA • DD ENERGODATA
11070 BEOGRAD, Lenjinov bulevar 12, P. fah 20, Tel (011) 138-520, 139-712, 222-3072


POVEZIVANJE U MREŽE POD NOVELL NETWARE-OM, INDUSTRIJSKE I LAB-KARTICE "ADVANTECH"
I OVLAŠĆENA PRODAJA FIRME OTC COMPUTER, TAIWAN
ComTec
PREDUZEĆE ZA INFORMATIČKI INŽENJERING I AUTOMATIZACIJU
NOVI SAD, VOJVODANSKIH BRIGADA 7/IX, ☎Fax 021 / 222-58

Isporuka SIGURNA

IMTEL U BEOGRADU

garancija 2 godine



MONITOR	MONOHROMATSKI VGA			KOLOR VGA		
HARD DISK / RAČUNAR	40MB	105 MB	210 MB	40MB	105 MB	210MB
486-33, 4MB RAMA, MINITOWER	4510	5180	5880	5030	5830	6530
386-40, 2MB RAMA, MINITOWER	3810	4480	5180	4460	5130	5830
386SX-25,2MB RAMA, STAND. KUĆIŠTE	2680	3350	4050	3330	4000	4700
286-16, 1MB RAMA, STAND. KUĆIŠTE	1880	2530	3250	2530	3200	3900

CENE SU U DM
I OBRAČUNAVAJU SE U DINARE PO KURSU MARKE NA SLOBODNOM TRŽIŠTU

ŠTAMPAČI	
HP LASER JET III P	3700
MEM. PROŠ. ZA LJ III P	400
TONER ZA LJ IIP	330
EPSON FX 1050	1450
EPSON LX 800	700

KOPROCESORI:	
AMD 80C287-12	180
80387SX 20	550

MODEMI 2400 BAUDA:	
POCKET	200
EKSTERNI MNP5	400

SOFTVER:	
DR DOS 4.3	100
MS DOS 5.0	250
MS WINDOWS 3.0	350

EPROM PROGRAMER	400
HP-IB KARTICA	550
AD/DA KARTICA	950
STRIMER COLORADO 120MB	800
MIŠ GM-6	55
FLOPI 1.44MB	150

NOVELL NETWARE
MREŽNI OPERATIVNI SITEM

DISKETE
1.2 MB
1.44MB

imtel inside

Sajamski popust!

RAČUNARI SU JAČI KADA SU POVEZANI !



NOVELL NETWARE 386 V3.11 PODRŽAVA DO 100 KORISNIKA	16000 DM
NOVELL NETWARE 386 V3.11 PODRŽAVA DO 20 KORISNIKA	8000 DM
NOVELL NETWARE 286 V2.2 PODRŽAVA DO 50 KORISNIKA	8000 DM
NOVELL NETWARE 286 V2.2 PODRŽAVA DO 10 KORISNIKA	5000 DM
ETHERNET MREŽNA KARTICA NE2000	550 DM

Sajamski popust!

IMTEL - Institut za Mikrotalasnu Tehniku i Elektroniku

Bulevar Lenjina 165 b, 11070 Novi Beograd
Telefon: (011) 134-516, Fax: (011) 138-928, Telex: 12797 IMTELYU, Žiro račun 60815-603-14396



SHOW


EPSON
VELIKO
IME,
POUZDAN
KVALITET!
EPSON
COMPUTERS
IDAC Tuzla
ASYS COMMERCE Novi Sad
MIKRONOVA Novi Sad
INFOTRADE Priština







AB soft
PROGRAMSKI PAKETI :
Programi za robno, materijalno i finansijsko poslovanje, obračun poreza, plata, kamata, evidencija čekova, virmani, komercijala, vodjenje magacina, blagajne, maloprodaja, vodjenje turističke agencije, prodaja vozila, programi po narudžbi...
OBUKA PO DOGOVORU!
Tel/fax 011/647-480


GAMA
Electronics
POSTOJANO SA VAMA!